



FAKULTÄT FÜR  
INFORMATIK

# Untersuchung von Wechselwirkungen zwischen Schwärmen mit unterschiedlichen Verhalten und Eigenschaften

Masterarbeit  
von

Mike Mikuteit

Betreuer:  
Prof. Dr.-Ing. Sanaz Mostaghim

Vorgelegt am 21.07.2014

## **Danksagung**

Ich danke meiner gesamten Familie, die immer an mich geglaubt hat und durch ihre finanzielle Unterstützung mein Studium überhaupt erst ermöglicht haben. Ein großer Dank gilt auch allen meinen Freunden, die immer für mich da waren und auch während der Masterarbeit für den nötigen Spaß und Ablenkung gesorgt haben, die mich zum Weitermachen motiviert haben. Außerdem möchte ich mich noch bei Frau Prof. Dr.-Ing. Sanaz Mostaghim bedanken. Sie hat mir die Möglichkeit gegeben die Masterarbeit in diesem, für mich sehr interessanten Forschungsbereich anzufertigen und stand mir im Verlauf der Arbeit stets mit fachlich kompetenter Unterstützung zur Seite.

## **Erklärung der Selbstständigkeit**

Hiermit versichere ich, Mike Mikuteit, die vorliegende Masterarbeit allein und nur unter Verwendung der angegebenen Quellen angefertigt zu haben.

Magdeburg, den 21. Juli 2014

---

Mike Mikuteit

# Inhaltsverzeichnis

Abbildungsverzeichnis	5
Tabellenverzeichnis	6
Pseudocodeverzeichnis	7
<b>1 Einleitung</b>	<b>8</b>
1.1 Motivation	8
1.2 Zielsetzung	9
1.3 Gliederung	9
<b>2 Grundlagen</b>	<b>11</b>
2.1 Schwärme	11
2.2 Künstliche Intelligenz	12
2.3 Schwarmintelligenz	12
2.4 Regeln für Schwarmindividuen	14
<b>3 Neue Ansätze für die Betrachtung von Multi-Schwärmen</b>	<b>20</b>
3.1 Arten von Wechselwirkungen zwischen Schwärmen	20
3.1.1 Keine Wechselwirkungen	21
3.1.2 Wechselwirkungen zwischen Schwärmen mit identischen Eigenschaften	21
3.1.3 Wechselwirkungen zwischen Schwärmen mit unterschiedlichen Eigenschaften	22
3.2 Stucked Partikel	23
3.2.1 Alternierende Stucked Partikel	23
3.2.2 Random Move Stucked Partikel	24
3.3 Erkennung von Stucked Partikeln	25
3.3.1 Nachbarschafts Veränderungen	26
3.3.2 Positionsänderung des Partikels	26
3.3.3 Winkelmessung zwischen Bewegungsvektoren	28
3.4 Behandlung von Stucked Partikeln	29
3.4.1 Festhalten	29
3.4.2 Adaptive Radien	30
<b>4 Implementierung</b>	<b>32</b>
4.1 Verwendete Werkzeuge und Hilfsmittel	32
4.2 Programmiertechnische Umsetzung	32
4.3 Gestaltung der Nutzeroberfläche	39

4.3.1	Zweck und Verwendung der GUI-Elemente . . . . .	39
4.3.2	Ein erwähnenswertes verworfenes Gestaltungselemente der grafischen Benutzeroberfläche . . . . .	42
<b>5</b>	<b>Evaluierung</b>	<b>44</b>
5.1	Wechselwirkungen . . . . .	44
5.1.1	Musterentstehung . . . . .	49
5.2	Instabilität durch Wechselwirkungen . . . . .	56
5.2.1	Stucked Partikel . . . . .	56
5.2.2	Zufälliges Löschen von Partikeln . . . . .	57
5.3	Stabilisierung durch Behandlung von Stucked Partikeln . . . . .	60
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>64</b>
6.1	Mögliche weiterführende Ansätze . . . . .	64
6.2	Zusammenfassung . . . . .	66
	<b>Literaturverzeichnis</b>	<b>68</b>

# Abbildungsverzeichnis

2.1	Optimale Wegfindung von Ameisen . . . . .	14
2.2	Grafik zu Attraction und Repulsion . . . . .	15
2.3	Grafische Verdeutlichung der Bewegungsgleichung . . . . .	16
2.4	Begrenzte Attraction und Repulsion Funktion . . . . .	17
2.5	Illustration Repulsion- und Attractionradius . . . . .	18
2.6	Funktion für Attraction- und Repulsionradius . . . . .	19
2.7	Stabilisierung eines Schwarms . . . . .	19
3.1	Illustration eines alternierenden Stucked Partikels . . . . .	24
3.2	Illustration eines Random Move Stucked Partikels . . . . .	25
3.3	Winkel zwischen Bewegungsvektoren von Partikeln . . . . .	28
4.1	Grafische Benutzeroberfläche des Softwareprototypen . . . . .	40
4.2	Verworfen GUI zum hinzufügen eines Schwarms . . . . .	43
5.1	Resultat Simulation 1 . . . . .	46
5.2	Resultat Simulation 2 . . . . .	46
5.3	Resultat Simulation 3 . . . . .	47
5.4	Resultat Simulation 4 . . . . .	48
5.5	Resultat Simulation 5 . . . . .	49
5.6	Muster bei identischen Repulsion- und Attractionradien . . . . .	50
5.7	Standartabweichung bei Vollständiger Vermischung . . . . .	51
5.8	Muster bei aufsteigenden Attractionradien . . . . .	51
5.9	Standartabweichung beim vollkommen überlappenden Zwiebel- schalen Muster . . . . .	52
5.10	Muster bei sich überschneidenden Repulsion- und Attractionradien . . . . .	53
5.11	Standartabweichung beim paarweise disjunkt überlappenden Zwie- belschalen Muster . . . . .	54
5.12	Dichteverteilung beim paarweise disjunkt überlappenden Zwie- belschalen Muster . . . . .	55
5.13	Muster bei aufsteigenden Repulsion- und Attractionradien . . . . .	55
5.14	Standartabweichung beim disjunkten Zwiebelschalen Muster . . . . .	56
5.15	Dichteverteilung beim disjunkten Zwiebelschalen Muster . . . . .	57
5.16	Trajektorie eines Stucked Partikels . . . . .	58
5.17	Initialisierung und Resultat beim zufälligen Partikellöschen . . . . .	59
5.18	Stabilisierung ohne Behandlung . . . . .	61
5.19	Stabilisierung druch Festhalten . . . . .	61
5.20	Stabilisierung durch Radienanpassung mit festem Wert . . . . .	62

---

5.21	Stabilisierung durch prozentuale Radienanpassung . . . . .	62
5.22	Stabilisierung durch prozentuale Radienanpassung und Beschleunigung . . . . .	63

# Tabellenverzeichnis

5.1	Die Tabelle listet die Parameterkombinationen der Schwärme in den fünf verschiedenen Simulationsläufen zu den Wechselwirkungen von Schwärmen auf . . . . .	45
5.2	Die Tabelle listet die Parameterkombinationen der Schwärme in der Simulation zum zufälligen Löschen von Partikeln auf . . . . .	60
5.3	Die Tabelle listet die Parameterkombinationen der Schwärme in den Simulationen zur Stabilisierung durch Behandlung von Stucked Partikeln auf . . . . .	61

# Pseudocodeverzeichnis

1	Berechnung des Schwärmverhaltens . . . . .	33
2	Erkennung von alternierenden Stucked Partikeln durch Positionsänderungen . . . . .	34
3	Erkennung von Random Move Stucked Partikeln durch Positionsänderungen . . . . .	35
4	Erkennung von Stucked Partikeln durch Winkelmessung . . . . .	36
5	Behandlung von Stucked Partikeln . . . . .	38
6	Erweiterung zur Beschleunigung von Stucked Partikeln . . . . .	39

# Kapitel 1

## Einleitung

### 1.1 Motivation

Um die Forschung im Bereich der Schwarmintelligenz weiter voran zu treiben, ist es nicht nur wichtig das Verhalten eines einzelnen Schwarms und dessen zugehöriger Individuen zu analysieren. Auch die Untersuchung der Wechselwirkungen zwischen mehreren Schwärmen gewinnt immer mehr an Bedeutung. Diese Wechselwirkungen wurden allerdings bisher kaum erforscht.

In der heutigen Forschung werden Roboter, die zusammen einen Schwarm bilden sollen, jeweils als Individuen dieses Schwarms betrachtet. Auch, wenn diese Vorgehensweise im ersten Moment einleuchtend scheint, bringt sie dennoch Probleme mit sich. In der Realität sind nicht alle Roboter gleich, sondern unterscheiden sich geringfügig voneinander. So kann es beispielsweise sein, dass ein auf Rädern fahrender Roboter keine hundertprozentig gerade Achsenstellung besitzt und so beim Befehl geradeaus zu fahren, ganz leicht schräg fährt. Außerdem ist es möglich, dass sich die Roboter auf unebenen Terrain bewegen. Das könnte beispielsweise steiniger oder hügeliger Untergrund sein. Im Falle von fliegenden Robotern besteht die Möglichkeit, dass diese bei Witterungen zum Einsatz kommen sollen, bei denen keine hundertprozentige Windstille garantiert werden kann. So kann es dazu kommen, dass die fliegenden Roboter durch Luftturbulenzen von ihrer eigentlichen Position verdrängt werden. Diese Unterschiede zwischen den Robotern und die Vielfalt der möglichen Einsatzgebiete, führen in der Realität dazu, dass sich ein echter Roboterschwarm anders verhält, als ein simulierter Roboterschwarm. Die Unterschiede der einzelnen Roboter, ob durch die Beschaffenheit der Roboter selbst oder durch ihren Einsatzbereich hervorgerufen, könnten theoretisch auch als verschiedene Schärmeigenschaften angesehen werden und somit jeder Roboter als Teil eines anderen, eigenen Schwarms. Das bedeutet um bessere reale Ergebnisse zu erzielen, müsste man einen Schwarm aus vielen Schwärmen annehmen. Dieser Schwarm würde aus  $n$  Schwärmen mit jeweils nur einem Individuum bestehen. Für eine Umsetzung dieser Annahme in die Realität, ist ein Verständnis für die Wechselwirkungen zwischen Schwärmen unerlässlich.

Was geschieht wenn mehrere Individuen, die zu unterschiedlichen Schwärmen gehören, aufeinander treffen? Kann die Stabilität der einzelnen Schwärme noch gewährleistet werden? Ist es möglich das Individuen, die verschiedenen Schwärmen

angehören, diese Schwärme nach einiger Zeit stabil bilden können, wenn sich die Individuen anfangs in einer zufälligen und vermischten Lage befinden? Existieren Möglichkeiten um instabile Multi-Schwarmkonstellationen zu stabilisieren? Diese Arbeit unternimmt erste Ansätze diese Fragen zu beantworten.

## 1.2 Zielsetzung

Das Ziel dieser Masterarbeit ist, eine erste Untersuchung der Wechselwirkungen zwischen Schwärmen. Es werden auch mögliche, durch diese Wechselwirkungen auftretende, Probleme und verschiedene neu entwickelte Lösungsansätze, zum beheben dieser Probleme, aufgeführt. Hierbei werden grundsätzlich mögliche Arten von Wechselwirkungen erstmals definiert und beschrieben. Aus den Wechselwirkungen resultierende Probleme werden aufgeschlüsselt und konkret definiert. Des Weiteren werden mögliche Vorgehensweisen zum automatischen Erkennen und Beheben dieser Probleme, mit Hilfe von Algorithmen, dargelegt. Diese grundlegenden Überlegungen werden dann technisch, mit Hilfe der Programmierumgebung MatLab, umgesetzt. Dabei wird auch der *Prototyp*<sup>1</sup> einer *GUI*<sup>2</sup> entwickelt. Da die oben genannten Überlegungen mit Hilfe der technischen Umsetzung simuliert werden sollen, soll die entwickelte GUI der übersichtlicheren und schnelleren Eingabe der Simulationsparameter dienen. Die Simulationen werden dann ausgewertet und untersucht, wobei auch die durch sie neu gewonnene Erkenntnis geschildert und interpretiert werden. Es wird außerdem dargelegt, ob die neu entwickelten Ansätze hilfreich und zielführend sind, indem sie durch die Simulationen ausgetestet werden.

## 1.3 Gliederung

Diese Masterarbeit teilt sich in die nachfolgenden Kapitel ein:

- **Kapitel 2: Grundlagen**

Dieses Kapitel schafft ein Grundverständnis der für die Arbeit wichtigen Begriffe der Künstlichen Intelligenz und vor Allem für deren Teilbereich, der Schwarmintelligenz. Zusätzlich werde die für diese Arbeit essenziellen Regeln für das Verhalten von Individuen innerhalb eines Schwarms aufgezeigt, da diese Regeln letztendlich die Grundlage für alle Simulationen in dieser Arbeit bilden.

- **Kapitel 3: Neue Ansätze für die Betrachtung von Multi-Schwärmen**

Die grundsätzlichen Überlegungen zu den Prinzipien der Schwarmintelligenz sind Hauptthema dieses Kapitels. Die Arten der Wechselwirkungen zwischen Schwärmen sind genauso Teil des Kapitels, wie auch der erstmals für diese Arbeit eingeführte Begriff der „*Stucked Partikel*“. Dabei

---

<sup>1</sup>Prototyp: (in der Softwareentwicklung) ein lauffähiges Stück Software oder eine anderweitige konkrete Modellierung einer Teilkomponente des Zielsystems. Ein Prototyp dient oft als Basis für bessere Kommunikation z.B. innerhalb eines Entwicklungsteams über konkrete Dinge (statt abstrakter Modelle) [WiPro, 2014].

<sup>2</sup>GUI: Kurzform für Graphical User Interface zu deutsch grafische Benutzeroberfläche

werden diese Partikel nicht nur definiert, es werden auch Möglichkeiten sie zu erkennen und zu behandeln aufgeschlüsselt.

- **Kapitel 4: Implementierung**

Die technische Umsetzung, der für diese Arbeit nötigen Simulationsschritte und entwickelten Methoden, ist die Grundlage des Kapitels Implementierung. Es werden dabei die verwendeten Werkzeuge benannt und die benutzten Hilfsmittel aufgelistet. Im weiteren Verlauf des Kapitels wird auf die technische Umsetzung der für die Arbeit entwickelten Algorithmen eingegangen. Zusätzlich bildet die detaillierte Beschreibung der, spezielle für diese Arbeit erstellten, grafischen Benutzeroberfläche einen großen Teil dieses Kapitels.

- **Kapitel 5: Evaluierung**

In diesem Kapitel werden die aus der Arbeit gewonnen Ergebnisse und Erkenntnisse aufgeschlüsselt und analysiert. Dabei wird auf die Wechselwirkungen zwischen Schwärmen genauso eingegangen, wie auf die dabei entstehenden Muster. Das Vorhandensein dieser Muster wird zusätzlich mathematisch bewiesen. Auch die Auswertung, der beim Wechselwirken von Schwärmen entstehenden Instabilität bilden einen Teil des Kapitels. Außerdem wird Bezug auf die erfolgreich angewandten Methoden zur Stabilisierung von Schwärmen genommen.

- **Kapitel 6: Zusammenfassung und Ausblick**

Das abschließende Kapitel beschäftigt sich mit einigen möglichen weiterführenden Ansätzen, die sich aus den Erkenntnissen dieser Masterarbeit ergeben. Zusätzlich wird hier die gesamte Arbeit kurz zusammengefasst.

## Kapitel 2

# Grundlagen

Dieses Kapitel bietet einen kurzen Überblick über die verschiedenen Grundlagen im Bereich der Künstlichen Intelligenz und vor Allem im Teilbereich der Schwarmintelligenz, auf denen diese Arbeit aufbaut. Es werden die wichtigsten Grundbegriffe kurz veranschaulicht. Dazu wird anfangs der Begriff Schwärme näher erläutert und hauptsächlich Bezug darauf genommen wie dieser innerhalb der Masterarbeit aufgefasst wird. Im Anschluss wird allgemein auf den Begriff der Künstlichen Intelligenz in der Informatik eingegangen. Dieser bildet eine der Hauptgrundlagen der Arbeit. Der Grund dafür ist das diese Arbeit direkt auf den Begriff der Schwarmintelligenz aufbaut. Schwarmintelligenz stellt, wie Oben bereits erwähnt, ein Teilgebiet der Künstlichen Intelligenz dar. Sie wird im dritten Abschnitt des Kapitels näher beleuchtet und an einem Beispiel verdeutlicht. Am Ende dieses Kapitels werden detailliert die einfachen Regeln dargelegt, nach denen, die in dieser Arbeit betrachteten, Individuen eines Schwarms agieren und somit der Verhalten des Schwarms bestimmen.

### 2.1 Schwärme

Ein Schwarm bezeichnet einen Zusammenschluss vom mindestens drei, aber häufig weitaus mehr Individuen, zu einer großen Gruppe, welche gemeinschaftlich agiert. Für solche Zusammenschlüsse gibt es mannigfaltige Beispiele aus der Natur. Zwei der populärsten sind Vogelschwärme und Fischeschwärme. Aber auch viele andere Lebewesen bilden Schwärme. So kann beispielsweise eine Kuhherde ebenfalls als ein Schwarm angesehen werden. Auch beim Menschen kommt es zum Phänomen der Schwarmbildung. Man könnte menschliche Gesellschaften als Schwarm ansehen. Ein noch viel offensichtlicheres Beispiel der Schwarmbildung beim Menschen, lässt sich bei Massenveranstaltungen, wie Musikfestivals oder großen Sportveranstaltungen, beobachten. Beim betrachten von Luftaufnahmen solcher Veranstaltungen, lassen sich häufig gerichtete Bewegungen von sehr großen Gruppen hin zu einem bestimmten Ziel erkennen. Diese Menschengruppen agieren als Schwarm, welcher in eine Richtung drängt. Dabei ist dem einzelnen Menschen innerhalb dieser Gruppe, die aus der Luft gut zu erkennende Gesamtdynamik der Gruppe, nicht in seinen vollständigen Ausmaß bewusst.

Da Schwärme ein so weit verbreitetes Phänomen darstellen, sind sie auch für die Forschung in vielen unterschiedlichen Forschungsbereichen, von großem

Interesse. So gibt es diesbezüglich Forschungen im Bereich der Biologie, der Soziologie, der Physik, der Mathematik und der Informatik, um nur einige zu nennen. Diese Arbeit betrachte den Begriff des Schwarms hauptsächlich aus der Sicht der Informatik und Mathematik. So sind alle Schwarmindividuen, auf die in der Arbeit Bezug genommen wird mathematisch Masse - und Ausdehnung Lose Punkte. Außerdem wird für die Punkte der Begriff Partikel, anstelle des Begriffs Individuen verwendet, da dieser mehr Gegebenheiten und Eigenschaften von Punkten entspricht.

## 2.2 Künstliche Intelligenz

Als Künstliche Intelligenz wird das Teilgebiet der Informatik bezeichnet, welches sich mit der Erzeugung intelligenten Verhaltens, mit Hilfe eines Rechensystems beschäftigt. Wobei die künstliche Intelligenz selber zumeist im Rechenystem mit Zuhilfenahme von Computerprogrammen erzeugt wird. Diese Computerprogramme basieren wiederum auf, in beliebigen Programmiersprachen umgesetzten, Algorithmen, die intelligentes Verhalten künstlich erzeugen. Dabei kann es sein, dass manche dieser Algorithmen die Möglichkeit besitzen sich an, sich verändernde Situationen individuell anzupassen. Eine solche Anpassung kann beispielsweise durch die Fähigkeit der Algorithmen erzeugt werden, ihre Grundparameter in bestimmten Fällen abzuwandeln. Diese Fähigkeit verleiht den Algorithmen die Eigenschaft eine spezielle, auf eine bestimmte Situation ausgerichtete Verhaltensweise zu erlernen. Außerdem können sich manche dieser Algorithmen nicht nur individuell anpassen, sondern die erlernte Anpassung auch noch abspeichern. Dadurch erhalten sie die Fähigkeit aus einer Menge von bereits erlernten Anpassungen zurückzugreifen. So ist es den Algorithmen möglich auf neuen Situationen, die Ähnlichkeit mit bereits bekannten Situationen, an die sie sich bereits zuvor angepasst haben, mit bereits erlernten Verhalten zu reagieren. Natürlich stellt auch das Erkennen einer bereits bekannten Situation selbst eine intelligente Handlungsweise dar. Durch die Algorithmen wird also künstlich ein intelligentes Verhalten erzeugt. Es entsteht eine Künstliche Intelligenz.

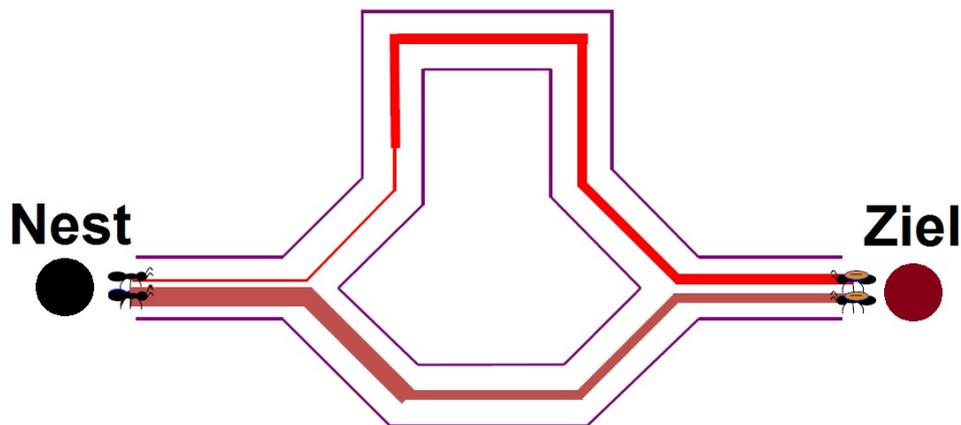
## 2.3 Schwarmintelligenz

Schwarmintelligenz ist ein Begriff der in vielen unterschiedlichen Forschungsbereichen Anwendung findet. So zum Beispiel in der Soziologie, der Biologie, der Mathematik und in der Informatik. Auch wenn die Übergänge zwischen den Bereichen bei der Schwarmintelligenzforschung zumeist fließend und häufig nicht eindeutig sind, versucht sich diese Arbeit vor Allem auf die Schwarmintelligenz im Bereich der Informatik zu beziehen. Hier versteht sich die Schwarmintelligenz als Teilbereich der Künstlichen Intelligenz. Doch um grundlegendes Verständnis für den Begriff zu schaffen muss dieser allgemein und nicht nur aus Sicht der Informatik dargelegt werden. In der Literatur sind häufig auch die Begriffe Gruppen- oder Kollektive Intelligenz als Synonyme für den Begriff der Schwarmintelligenz zu finden. Als Schwarmintelligenz wird allgemein das dezentralisierte Zusammenspiel von mehreren, sich selbst organisierenden, Individuen bezeichnet. Durch welches Abläufe, Resultate oder Verhaltensmuster erzeugt werden die, aus menschlicher Sicht, als intelligent interpretiert werden

können. [Beni et al., 1989] So entsteht durch das Zusammenwirken der Individuen eines Schwarms eine Art Klugheit des Kollektivs. Die Zusammenarbeit der Individuen dient dabei der Erreichung eines bestimmten Zieles. Dieses Ziel muss allerdings nicht zwangsläufig jedem Individuum bewusst sein. Im Gegenteil sogar, häufig ist keinem Individuum des Schwarms das erzeugte Ergebnis bekannt. Die Zusammenarbeit als Schwarm bietet allerdings meist jedem Individuum, welches sich darin befindet, einen persönlichen Vorteil. Dieser kann zum Beispiel individuelle Aufwandsverringerung, durch Arbeitsteilung oder Schutz, durch die Größe des Schwarms sein. Hierin liegt der Grund aus dem sich mehrere Individuen zu Schwärmen vereinen.

Der Begriff der Schwarmintelligenz lässt sich sehr gut am Beispiel einer Ameisenkolonie verdeutlichen. [RoMa, 2007] Ameisen erschaffen durch das Zusammenwirken sehr vieler Individuen große und komplexe Strukturen, wie z.B. Nester oder Ameisenstraßen. Die einzelne Ameise ist sich dabei der Dimension des Erschaffenen keines Wegs bewusst. [Spiteller 2013] So sind Ameisen in der Lage optimale Wege von ihrem Nest zu einem Ziel zu finden. Das Ziel könnte zum Beispiel Nahrung oder für den Weiterbau des Nestes benötigte Materialien sein. Die Ameisen finden dabei den optimalen Weg mit Hilfe einer Pheromonspur, welche umso stärker wird, umso mehr Ameisen ein und denselben Weg ablaufen sind. Die Pheromonspur schnell am stärksten, da die Ameisen hier den Weg zwischen Nest und Ziel und wieder zurück, am schnellsten zurücklegen können. Irgendwann ist die Pheromonspur auf dem optimalen Weg so stark, dass alle Ameisen nur diesen Weg wählen. Keine der Ameisen weiß bei diesem Vorgehen vorher, welcher der optimale Weg ist, noch wissen sie es, wenn dieser Weg gefunden ist. Jede einzelne Ameise folgt ausschließlich der stärksten Pheromonspur. Abbildung 2.1 verdeutlicht skizzenhaft das Vorgehen der Ameisen. Diese Art der optimalen Wegfindung ist so effizient, dass sich im Bereich der Optimierung die Klasse der Ameisenalgorithmen entwickelt hat. [Dorigo et al. 2000] Ameisenalgorithmen zur Optimierung basieren allesamt grundlegend auf der oben beschriebenen Methode der optimalen Wegfindung von Ameisen.

Ein weiteres Beispiel zur Verdeutlichung von Schwarmintelligenz, sind die sogenannten „*Desire Lines*“. Hierbei handelt es sich um die Umgestaltung von Trampelpfaden zu befestigten Wegen, um so neuen Wege optimal anzulegen. Dieses Vorgehen beruht auf einem Experiment des Architekten Christopher Alexander [Alexander 1975]. Er wurde in den siebziger Jahren von der Leitung der Universität von Oregon zur Umgestaltung ihres Universitätscampus engagiert. Alexander entschied sich dafür das Gelände zwischen den Universitätsgebäuden vollständig zu planieren und dort Rasen aussähen zu lassen. Anschließend ließ man den Universitätsbetrieb wie üblich weiterlaufen, allerdings konnten die einzelnen Gebäude auf dem Campus nur noch durch Überqueren der Rasenfläche erreicht werden. Nach dreimonatiger Wartezeit entstanden so Trampelpfade auf der Rasenfläche. Diese Trampelpfade wurden dann befestigt. So konnte man Wege an den, für Studenten und Mitarbeiter, optimalen Positionen anlegen. Die Wege wurden also nicht, wie üblich, durch die direkte Planung eines Architekten festgelegt. Sondern sie bildeten sich durch das, für jeden einzelnen eher unbewusste, Zusammenwirken aller am Universitätsalltag beteiligten Menschen heraus. Indem jeder den für sich optimalen Weg auswählte, wurden die bestmöglichen Wegpositionen automatisch nach einiger Zeit durch die Entstehung der Trampelpfade deutlich. Dies ist ein Ergebnis, dessen Lösung ein gewisses Maß an



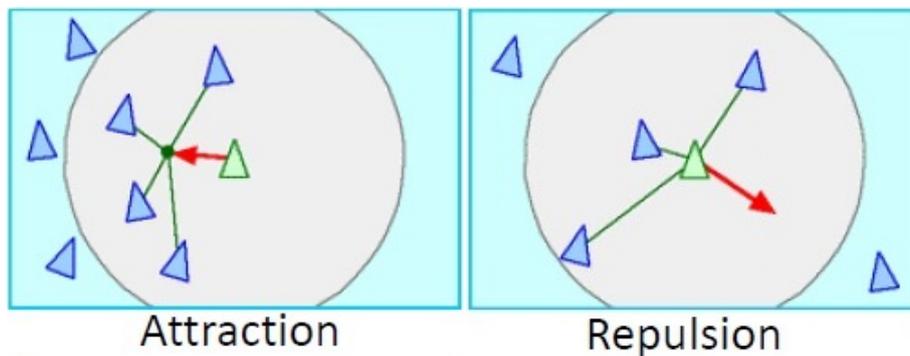
**Abbildung 2.1:** Die Ameisen haben zwei mögliche Wege vom Nest zum Ziel und wieder zurück. Der untere ist der Optimale. Die Pheromonspur wird dort schneller wesentlich stärker, als beim oberen Weg, da die Strecke schneller zurückgelegt werden kann [Mostaghim, 2013]

Intelligenz erfordert. Die nötige Intelligenz zur Lösung des Problems der besten Wegpositionen auf dem Campus, wurde hier allerdings von der gesamten Gruppe und nicht von einem einzelnen Individuum hervorgebracht. Dieses Experiment ist ein perfektes Beispiel dafür, dass Schwarmintelligenz auch beim Menschen auftritt [Bryant 1991].

Die oben aufgeschlüsselten Beispiele machen deutlich, dass Schwarmintelligenz ein in der Natur weit verbreitetes Phänomen darstellt. Die einzelnen Individuen folgen dabei zumeist nur sehr einfachen Regeln. So zum Beispiel das Verfolgen der stärksten Pheromonspur, wie im Ameisenbeispiel oder das Wählen des für das Individuum selbst besten Weges, wie im „*Desire Lines*“-Beispiel. Im nachfolgenden Kapitel werden die Regeln veranschaulicht, welche das Verhalten, der in dieser Masterarbeit verwendeten Individuen (Partikel) hervorruft.

## 2.4 Regeln für Schwarmindividuen

Die in dieser Arbeit betrachteten Schwärme, bestehen aus Individuen, welche alle den selben einfachen Grundregeln folgen. Wobei die Regeln für alle Individuen vollkommen identisch sind, solange die Individuen zu ein und demselben Schwarm gehören. Gehören die Individuen verschiedenen Schwärmen an, so unterscheiden sich im allgemeinen auch die Regeln, denen sie folgen, voneinander auch wenn die Grundprinzipien, im Falle dieser Arbeit, immer gleich bleiben. Wie bereits in den Vorangehenden Abschnitten erwähnt sind die Individuen, die in dieser Arbeit betrachtet werden ausschließlich mathematisch Ausdehnung- und Masse lose Punkte. Diese werden hier als Partikel bezeichnet. Jeder der in der Arbeit betrachteten Partikel besitzt grundlegend genau zwei simple Steuerungsverhaltensweisen, die seine Bewegungen verursachen. Diese beziehen sich beide auf die Partikel, die in der Nachbarschaft positioniert sind. Die Steuerungsverhaltensweisen basieren auf zwei, der drei erstmals von [Reynolds, 1987] eingeführten Verhaltensweisen. Diese sind die Separation, die Kohäsion und die Angleichung.



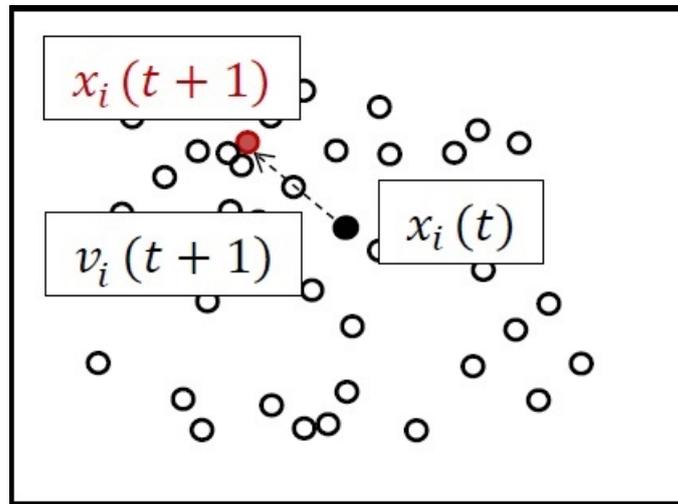
**Abbildung 2.2:** Auf der linken Abbildung wirken Kohäsionskräfte zwischen **grünen Dreieck** und seinen **Nachbarn**, es bewegt sich dadurch auf sie zu (*Attraction*). In der rechten Abbildung wirken Separationskräfte zwischen dem **grünen Dreieck** und seinen **Nachbarn**, es bewegt sich dadurch von ihnen weg (*Repulsion*). [Reynolds, 1999]

Da nur Separation und Kohäsion in dieser Arbeit Anwendung finden, wird die Angleichung als dritte Steuerverhaltensweise hier nicht erläutert.

Bei der Kohäsion wird ein Partikel des Schwarms von anderen Partikeln angezogen und umgekehrt. Der Partikel bewegt sich dadurch auf die anderen Partikel zu und ihre Abstände zueinander verringern sich. Es liegt die sogenannte *Attraction*, also Anziehung vor. Bei der Separation wird ein Partikel des Schwarms von anderen Partikeln abgestoßen und umgekehrt. Der Partikel bewegt sich dadurch von den anderen Partikel weg und ihre Abstände zueinander vergrößern sich. Es liegt die sogenannte *Repulsion*, also Abstoßung vor. In Abbildung 2.2 wird das *Attraction* und *Repulsion* Verhalten skizzenhaft verdeutlicht.

In der Abbildung ist gut zu erkennen wie sich ein Partikel jeweils im Falle von *Attraction* oder *Repulsion* verhält. So sieht man in der linken Darstellung wie sich der grüne Partikel auf den Massenschwerpunkt zwischen seinen vier nächsten Nachbarpartikeln zu bewegt, da hier der Abstand zu allen vier Nachbarn am geringsten ist. Hier herrscht also *Attraction* vor. In der rechten Skizze erkennt man wie sich der grüne Partikel von seinen drei nächsten Nachbarpartikeln weg bewegt, um den Abstand zwischen sich selbst und den Nachbarn zu erhöhen. Er herrscht *Repulsion* vor.

Mathematisch gesehen stellt sich die Positionsänderung des Partikels wie folgt dar. Die Position des Schwarmpartikels  $i$  zum Zeitpunkt  $t$  wird beschrieben durch  $x_i(t) \in \mathbf{R}^n$ . Zum darauffolgenden Zeitpunkt  $t + 1$  befindet sich Partikel  $i$  an der Position  $x_i(t + 1)$ . Diese Position stellt sich wie folgt dar  $x_i(t + 1) = x_i(t) + v_i(t + 1)$ . Wobei  $v_i(t + 1) = wv_i(t) + f_i$ . Dies wird als Bewegungsgleichung bezeichnet. Abbildung 2.3 verdeutlicht die Positionsänderung mit Hilfe diese Bewegungsgleichung noch einmal grafisch. Innerhalb der Gleichung stellt  $v_i$  die Geschwindigkeit des Partikels  $i$  dar und  $w$  einen Skalierungsfaktor, der bestimmt mit welcher Gewichtung die alte Geschwindigkeit von  $i$ , also  $v_i(t)$ , in die Berechnung der neuen Position  $x_i(t + 1)$  eingeht.  $f_i$  bezeichnet die verwendete Funktion. Die Funktion ist beliebig wählbar und erzeugt je nach Definition das erwünschte Verhalten der Partikel. Eine der einfachsten hier möglichen Funktionen, wäre eine lineare *Attraction*. Diese sorgt dafür das die

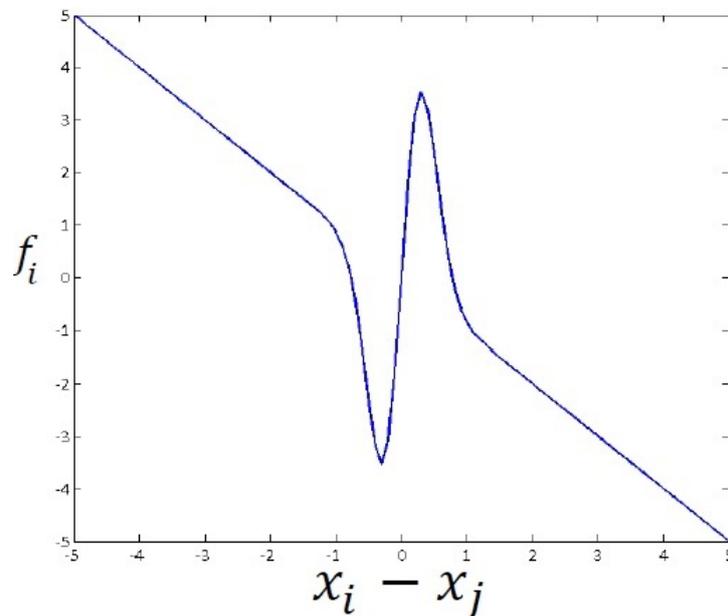


**Abbildung 2.3:** Partikel  $i$  befindet sich zum Zeitpunkt  $t$  an Position  $x_i(t)$  und bewegt sich durch Addition von  $v_i(t+1)$ , zum Zeitpunkt  $t+1$ , zur Position  $x_i(t+1)$  [Mostaghim, 2013]

Kohäsionskräfte zwischen den Partikeln umso stärker sind, umso weiter die Partikel voneinander entfernt sind. Verringert sich der Abstand, verringern sich auch die Kohäsionskräfte. Die Partikel bewegen sich also umso schneller aufeinander zu, je weiter sie voneinander entfernt sind und umso langsamer, je näher sie sich kommen. Die Funktion stellt sich mathematisch als  $f_i(i, j) = -k_a(x_i - x_j)$ ,  $k_a > 0$  dar. Der Werte  $x_i$  und  $x_j$  sind dabei die Positionen der Partikel  $i$  und  $j$ .  $k_a$  ist hier wieder ein Skalierungsfaktor, der die Geschwindigkeit regelt. Durch Invertierung des Vorzeichens entsteht aus der linearen *Attraction* eine lineare *Repulsion*, die ein genau umgekehrtes Verhalten der Partikel verursacht.

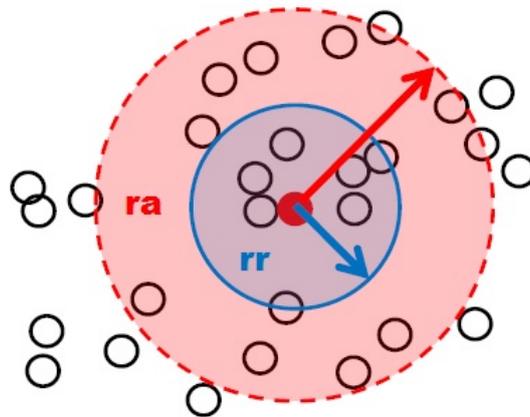
Es gibt noch viele weitere solcher Funktionen, die alle unterschiedliche Verhalten der Partikel auslösen. In dieser Arbeit wird die begrenzte *Attraction* und *Repulsion* Funktion genutzt um das Schwärmerverhalten, der Partikel innerhalb der Simulationen, zu erzeugen. Mathematisch lässt sich durch  $f_a = a$  für die *Attraction* und  $f_r = -b \exp(-\frac{\|x_i - x_j\|^2}{c})$  für die *Repulsion* beschreiben. Wobei  $a, b$  und  $c$  frei wählbare positive Konstanten sind,  $x_i$  die Position des Partikels  $i$  zum Zeitpunkt  $t$  und  $x_j$  die Position des Partikels  $j$  zum Zeitpunkt  $t$ . Die Abbildung 2.4 veranschaulicht die Werte der Funktion in Bezug auf den Abstand der Positionen, der Partikel  $i$  und  $j$ .

Die in dieser Arbeit betrachteten Schwärme sind keine vollständig verbundenen Netzwerke. Das bedeutet nicht jeder Partikel der zu einem Schwarm gehört, kann uneingeschränkt mit jedem anderen Partikel des Schwarms kommunizieren und so Informationen austauschen. Auch wenn eine vollständige Vernetzung zwischen allen Partikeln eines Schwarms mit den in der Arbeit genutzten Simulationswerkzeugen möglich wäre, wurde darauf hier bewusst verzichtet. Der Grund hierfür liegt in der grundlegenden Bezugnahme der Arbeit auf reale Systeme. So haben Individuen in realen Schwärmen nur einen begrenzten Kommunikationsradius. Dieser Fakt wird schnell deutlich, wenn man beispielsweise



**Abbildung 2.4:** Wert der Begrenzten Attraction und Repulsion Funktion in Bezug auf den Abstand zwischen den Partikeln  $i$  und  $j$  [Mostaghim, 2013]

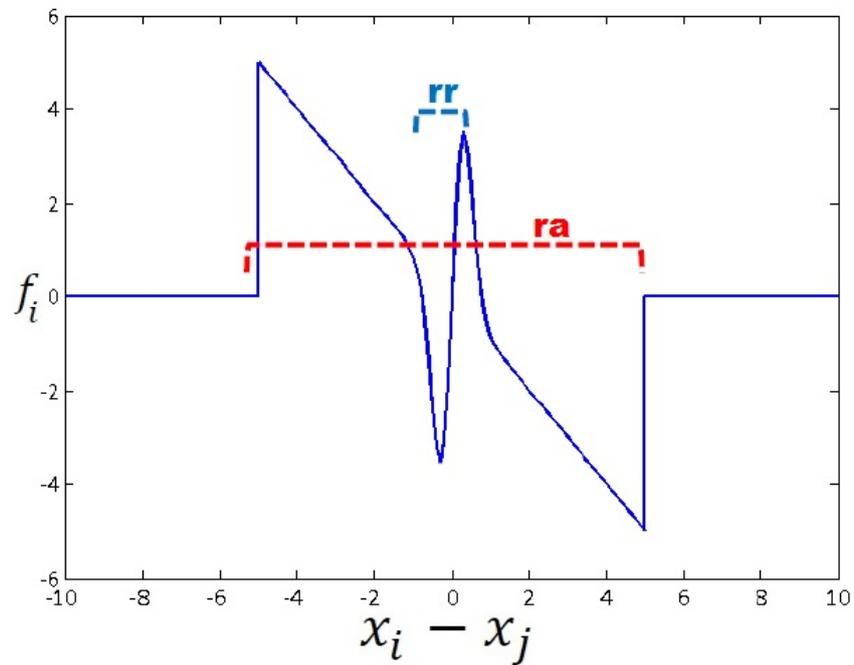
se die eigene Situation innerhalb einer großen Menschenmenge betrachtet. Befindet man sich, zum Beispiel bei Massenveranstaltungen, innerhalb einer solchen Menschenmenge, kann man nur mit den Menschen in unmittelbarer Nähe kommunizieren. Man besitzt also nur einen begrenzten Kommunikationsradius. Solch ein begrenzter Kommunikationsradius wird auch für die Partikel angenommen, welche die Schwärme in dieser Arbeit erzeugen. Jeder Partikel kann ausschließlich mit anderen Partikeln kommunizieren, die sich innerhalb seines Kommunikationsradius befinden. Unter Hinzunahme der oben Beschriebenen begrenzten Attraction und Repulsion Funktion für das Schwärmverhalten der Partikel, entstehen innerhalb des Kommunikationsradius jedes Partikels zwei Unterradien. Diese Unterradien sind der Repulsionradius und der Attractionradius. Der Repulsionradius ist kleiner als der Attractionradius und befindet sich direkt um den Partikel herum. Der Partikel stellt den Mittelpunkt dar. Alle Nachbarpartikel, die sich innerhalb dieses Radius befinden bewegen sich vom Partikel weg. Innerhalb dieses Radius wirkt die Separationskraft. Der Attractionradius ist größer als der Repulsionradius und schließt den Repulsionradius aus. Das bedeutet der Attractionradius liegt um den Repulsionradius herum. Er hat ebenfalls den Partikel selbst als Mittelpunkt. Innerhalb des Attractionradius herrschen Kohäsionskräfte. Das bedeutet, dass alle anderen Partikel, welche sich innerhalb dieses Radius befinden vom Partikel angezogen werden. Der sich aus Repulsion- und Attractionradius zusammensetzende Kommunikationsradius ist in Abbildung 2.5 skizziert. Da der Kommunikationsradius begrenzt ist entsteht daraus im Resultat auch eine begrenzte Funktion. Diese ist in Abbildung 2.6 dargestellt. Hier kann man sehr gut den abrupten Wertwechsel der Funktion auf null, am linken und rechten Rand, also bei zu großem Abstand der zweier Parti-



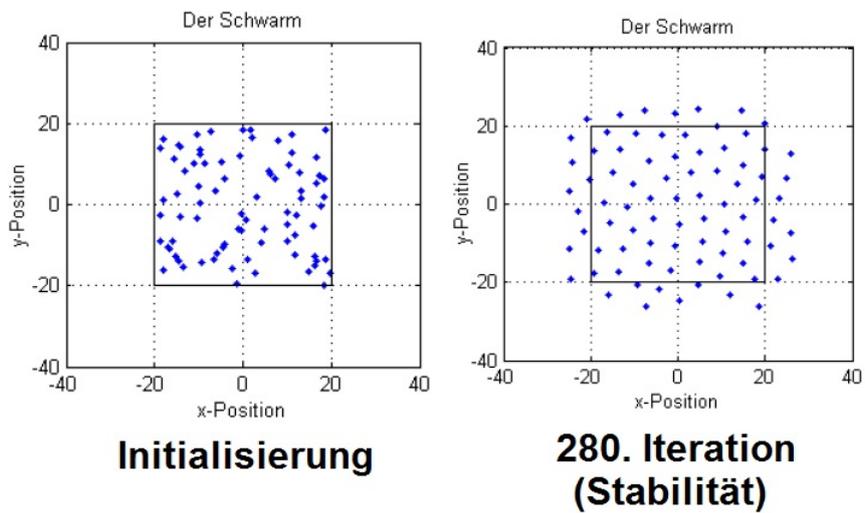
**Abbildung 2.5:** Der Kommunikationsradius des hervorgehobenen Partikels teilt sich in **Repulsionradius**( $rr$ ) und **Attractionradius** ( $ra$ ). Der Partikel kann ausschließlich mit anderen Partikeln innerhalb seines Kommunikationsradius Informationen austauschen [Mostaghim, 2013]

kel voreinander erkennen. Die beiden lokalen Extremstellen um die Wendestelle im Mittelteil der Funktion definieren den Repulsionradius, wobei der Rest der Funktion den Attractionradius um den Partikel  $i$  festlegt.

Ein Schwarm in den jedem Partikel genau diese Funktion als Verhaltensregel zugrunde liegt verhält sich wie in [Blum, 2013] beschrieben. Die Partikel suchen sich in einer endlichen Anzahl von Zeitschritten eine Position, welche ihnen einen optimalen Abstand zueinander, im Sinne der Funktion, garantiert. Haben alle Partikel diesen optimalen Abstand zueinander erreicht findet keine weitere Bewegung mehr statt. Die Partikel verharren an ihren Position und der Schwarm ist stabil. Dieses Phänomen ist in Abbildung 2.7 verdeutlicht. Die Partikel werden in zufälliger Lage initialisiert und haben sich in diesem Fall nach 280 Iterationen in eine für sie optimale Position zueinander bewegt und Stabilität erreicht. Da diese Verhaltensweisen einzelner Schwärme bereits intensiv in vorangehenden Arbeit, wie [Blum, 2013] beleuchtet wurden, wird es in den nachfolgenden Kapiteln dieser Arbeit vor Allem um die Betrachtung mehrerer unterschiedlicher Schwärme gehen.



**Abbildung 2.6:** Die begrenzte Attraction und Repulsion Funktion aufgeteilt zwischen **Repulsionradius(rr)** und **Attractionradius (ra)** innerhalb des Kommunikationsradius [Mostaghim, 2013]



**Abbildung 2.7:** Bei der Initialisierung liegen die Partikel in allgemeiner, zufälliger Lage zueinander. In diesem Beispiel haben sie sich nach 280 Iterationen in eine vollkommen stabile Position bewegt [Blum, 2013]

## Kapitel 3

# Neue Ansätze für die Betrachtung von Multi-Schwärmen

Da die Forschung im Bereich der Schwarmintelligenz einen noch relativ jungen Forschungszweig darstellt [Beni et al., 1989] [Hinchey et al., 2007], sind auch viele Teilfelder der Schwarmintelligenz bisher nur sehr wenig, bis gar nicht erforscht. So auch der Bereich der Wechselwirkungen zwischen Schwärmen. In diesem Kapitel werden die neuen speziell für diese Arbeit entwickelten Ansätze ausführlich erläutert. Zusätzlich werden die, aus der Betrachtung der Wechselwirkungen zwischen Schwärmen, entstehende Probleme detailliert geschildert. Dabei wird auf mögliche Arten von Wechselwirkungen zwischen Schwärmen eingegangen. Das, durch in Wechselwirkung tretende Schwärme mit unterschiedlichen Eigenschaften, entstehende Problem der Stucked Partikel, welche die Stabilisierung der Schwärme verhindern, wird erstmals definiert. Es werden mögliche Arten von Stucked Partikeln beschrieben. Des Weiteren werden Optionen zur automatischen Erkennung von Stucked Partikeln aufgeführt. Im Abschluss des Kapitels werden Möglichkeiten aufgeführt, mit denen man dem Problem der Stucked Partikel entgegenwirken kann, um so wechselwirkende Schwärme zu stabilisieren.

### 3.1 Arten von Wechselwirkungen zwischen Schwärmen

Wenn zwei Schwärme aufeinander treffen, treten diese für gewöhnlich miteinander in Wechselwirkung. Diese Wechselwirkungen sind bisher kaum untersucht, geschweige denn definiert wurden. In diesem Abschnitt sind drei mögliche Wechselwirkungen für unterschiedliche Fälle von aufeinandertreffenden Schwärmen beschrieben. Diese Fälle sind keine Wechselwirkungen, Wechselwirkungen zwischen Schwärmen mit identischen Eigenschaften und Wechselwirkungen zwischen Schwärmen mit unterschiedlichen Eigenschaften. Wobei auf den letzten Fall am intensivsten eingegangen wird, da hierauf der Hauptfokus der Arbeit liegt.

### 3.1.1 Keine Wechselwirkungen

Das Vorhandensein von keinerlei Wechselwirkungen zwischen Schwärmen, beschreibt den trivialen Fall. Dieser soll hier, aufgrund der erstmaligen Aufschlüsselung von möglichen Arten von Wechselwirkungen zwischen Schwärmen, trotzdem kurz Erwähnung finden. Bei dieser Art von Wechselwirkungen treffen zwei oder mehr Schwärme aufeinander und beeinflussen sich gegenseitig nicht. Es ist hierbei vollkommen unbedeutend ob die Schwärme identische oder unterschiedliche Eigenschaften besitzen, da dies auf diese Art von Wechselwirkungen keinerlei Auswirkung hat.

In einer Simulation von Partikeln die vollkommen durchlässig sind und so nicht miteinander kollidieren können, würde das bedeuten, dass die Schwärme wie in [Blum, 2013] beschrieben vollkommen stabil bleiben. Dieses Ergebnis tritt ein, da in diesem Fall weder Partikel innerhalb der Schwärme ausfallen, noch die Partikel eines Schwarmes, durch Partikel eines anderen Schwarmes, beeinflusst werden.

Man könnte allerdings auch den näher an der Realität angelehnten Fall, der nicht durchlässigen Partikel, annehmen. Hierbei kann es zu Kollisionen zwischen den Partikeln unterschiedlicher Schwärme kommen. Die Schwärme würden sich denn nach dem ebenfalls in [Blum, 2013] erläuterten Muster verhalten und sich minimal neu ausrichten, um sich zu stabilisieren. Allerdings muss bedacht werden, dass die minimalen Anpassungen zu erneuten Kollisionen führen könnten. Dies könnte im negativsten Falle eine Kettenreaktion auslösen, die unter Umständen den gesamten Schwarm durch Kollisionen auslöschen kann.

Vor Allem der zweite Fall macht deutlich, dass die grundsätzliche Annahme dieser Art von Wechselwirkung in möglichen realweltlichen Anwendungen, wie zum Beispiel fliegende Roboterschwärme, vermutlich nicht praktikabel sein dürfte. Das Risiko von Kollisionen und die damit verbundene Verschwendung von Ressourcen, wären wahrscheinlich zu hoch um ein Projekt unter dieser Annahme anzugehen. Jedoch ist nicht auszuschließen, dass in Zukunft computergestützte Problemlöseverfahren entwickelt werden, die genau diese Art von Wechselwirkungen zwischen Schwärmen annehmen. Hier wären die Ressourceneinbußen, wenn überhaupt nur minimal, da Kollisionen und damit einhergehende Ausfälle, nur hypothetisch als Teil einer Berechnung auftreten. Ungeachtet eventueller zukünftiger Entwicklungen, soll allerdings diese Art der Wechselwirkung zwischen Schwärmen nicht weiter Gegenstand der Betrachtungen in dieser Arbeit sein.

### 3.1.2 Wechselwirkungen zwischen Schwärmen mit identischen Eigenschaften

Es könnte in seltenen Fällen geschehen, dass Schwärme mit vollkommen identischen Eigenschaften aufeinander treffen und in Wechselwirkung treten. Das bedeutet alle Partikel innerhalb jedes Schwarmes bewegen sich nach den gleichen Verhaltensregeln. Kommt es zum Aufeinandertreffen zweier oder mehrerer Partikel, welche jeweils einem anderen Schwarm zugeordnet sind, wechselwirken diese wie zwei oder mehrere Partikel des selben Schwarmes, da kein Unterschied in ihren Eigenschaften besteht. So kann man beim Zusammentreffen mehrerer Schwärme mit vollkommen identischen Eigenschaften diese, bezogen auf das Verhalten beim Wechselwirken miteinander, als einen einzigen großen

Schwarm ansehen. Die Partikel der Schwärme werden sich, nach ihrer Initialisierung in zufälliger Lage, nach einer endlichen Anzahl von Zeitschritten in eine vollständig stabile Position bewegen und von diesem Zeitpunkt an, ohne das Einwirken äußerer Einflüsse, keine weiteren Bewegungen mehr ausführen. Es tritt das in [Blum, 2013] ausführlich untersuchte Verhalten eines einzigen Schwarms auf. Zwar sind die Partikel der einzelnen Schwärme, je nach Initialisierung, am Ende zufällig miteinander vermischt im Raum verteilt. Sie sind jedoch nach einer endlichen Anzahl von Iterationen definitiv stabil, da sie sich zusammen wie ein einziger Schwarm verhalten.

### 3.1.3 Wechselwirkungen zwischen Schwärmen mit unterschiedlichen Eigenschaften

Diese Wechselwirkungen treten auf wenn sich zwei oder mehr Schwärme mit unterschiedlichen Eigenschaften begegnen. Dabei gibt es zwei mögliche Szenarien. Zum Einen könnten die Partikel eines oder mehrere Schwärme fest im Raum verankert sein, während Schwärme mit frei beweglichen Partikeln auf auf dieser feststehenden Schwärme treffen. Zum Anderen könnten ausschließlich Schwärme mit frei beweglichen Partikeln in Wechselwirkung treten. Dieser zweite Fall wird in der Arbeit hauptsächlich betrachtet. Im Folgenden werden beide Fälle detailliert beschrieben:

- **Zusammentreffen von fest verankerten und frei beweglichen Schwärmen:**

Zum besseren Verständnis wird die entstehende Wechselwirkung hier nur anhand von zwei Schwärmen erläutert. Die beiden Schwärme haben unterschiedliche Eigenschaften und einer der Schwärme ist vollkommen feststehen, wobei der andere Schwarm frei bewegliche Partikel besitzt. Beim Zusammentreffen zweier solcher Schwärme, werden die Partikel des frei beweglichen Schwarms immer versuchen sich in Positionen innerhalb oder um den unbeweglichen Schwarm herum zu bewegen, die für diese Partikel Stabilität gemäß ihrer Eigenschaften garantieren. Eine vollständige Stabilisierung der Partikel nach einer endlichen Anzahl von Zeitschritten ist hier leider nicht in allen Fällen möglich. Der Grund hierfür liegt in den unendlich vielen Möglichkeiten der initialen Positionierung der Partikel. So kann es vorkommen, das ein Partikel des beweglichen Schwarms inmitten einer Menge von Partikeln des unbeweglichen Schwarms initialisiert wird. Die Partikel des unbeweglichen könnten dabei so nah am beweglichen Partikel und zueinander positioniert sein, dass es dem beweglichen Partikel nicht möglich ist eine stabile Position einzunehmen, die seinen Eigenschaften entspricht. Der bewegliche Partikel kann nicht aus der ihn umschließenden Partikelmenge heraus gelangen, weil diese Partikel zu nah beieinander angeordnet sind. Außerdem kann er inmitten der feststehenden Partikel keine stabile Position finden, da die unbeweglichen Partikel zu dicht um ihn positioniert sind. Aus diesem Grund wird sich der eingeschlossene Partikel des beweglichen Schwarms endlos weiter bewegen, ohne jemals eine stabile Position einnehmen zu können.

- **Zusammentreffen frei beweglicher Schwärme:**

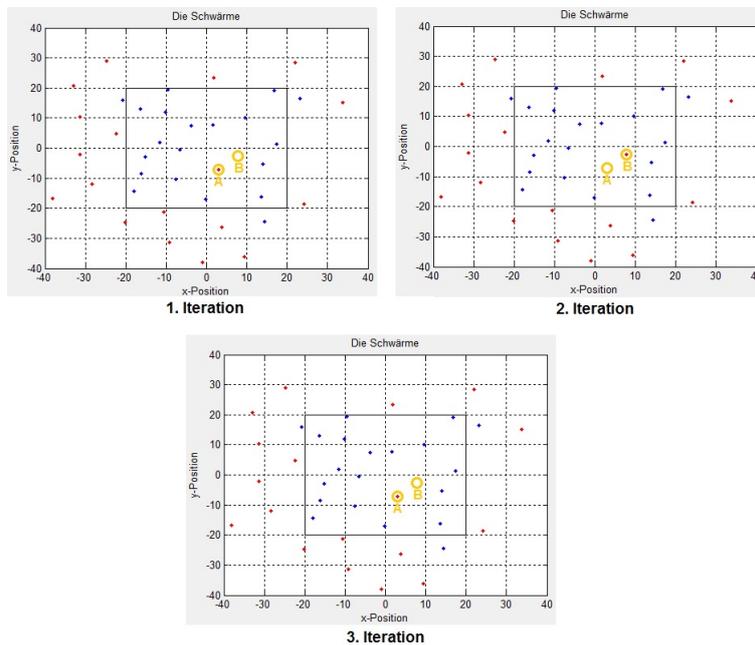
Auch hier wird der Sachverhalt zur Vereinfachung nur am Beispiel von zwei aufeinander treffenden Schwärmen dargelegt. Die Eigenschaften des einen Schwarms unterschieden sich von den Eigenschaften des anderen Schwarms und beide Schwärme besitzen vollkommen frei bewegliche Partikel. Treffen zwei solche Schwärme aufeinander bewegen sie sich so lange durcheinander, bis die meisten Partikel der zwei Schwärme eine Position eingenommen, die entsprechend ihrer eigenen Eigenschaft und in Bezug auf die Eigenschaft der Partikel des jeweils anderen Schwarms stabil ist. Eine hundert Prozentige Stabilität nach einer endlichen Anzahl von Zeitschritten, wie in [Blum, 2013] für einen Schwarm beschrieben, kann in diesem Fall jedoch leider nicht garantiert werden. Der Grund hierfür liegt darin, dass es unter Umständen geschehen kann das sich die Partikel eines Schwarms stabil um einen oder mehrere Partikel des anderen Schwarms anordnen. Die umschlossenen Partikel könnten jedoch Eigenschaften besitzen, die dazu führen, dass sich diese Partikel weiterbewegen, um Stabilität zu erreichen. Allerdings ist es ihnen durch ihre Eigenschaften nicht mehr möglich den Bereich der sie umschließenden Partikel zu verlassen. Die Anordnung der umschließenden Partikel verhindert jedoch eine Stabilisierung, die den Eigenschaft der umschlossenen Partikel gerecht wird. So könnten die umschließenden Partikel beispielsweise zu nah um die umschlossenen Partikel angeordnet sein. Die umschließenden Partikel befinden sich dabei in einer für sie so stabilen Lage bezüglich ihrer restlichen Umgebung, dass sie nicht mehr von den Eigenschaften der umschlossenen Partikel beeinflusst werden können. Diese Tatsache führt dazu, dass sich die umschlossenen Partikel im umschlossenen Gebiet endlos von einer Position auf die nächste bewegen ohne jemals Stabilität erreichen zu können. Dieses Phänomen der instabilen Partikel wird als Stucked Partikel bezeichnet und kann bei jeglicher Wechselwirkung zwischen Schwärmen mit unterschiedlichen Eigenschaften auftreten. Im nachfolgenden Abschnitt wird es detailliert betrachtet.

## 3.2 Stucked Partikel

Der Begriff des Stucked Partikels wurde speziell für diese Arbeit neu eingeführt, da dieses Phänomen sehr häufig auftritt, wenn Schwärme mit unterschiedlichen Eigenschaften in Wechselwirkung treten. Eine Stucked Partikel bezeichnet einen Partikel, der nach einer endlichen Anzahl  $n$  von Iterationen nicht stabil wird, sondern beginnt, innerhalb eines bestimmten Gebietes, mit jeder weiteren Iteration die Position zu wechseln. Das Phänomen der Stucked Partikel verhindert die garantierte Stabilisierung von Schwärmen, wenn zwei oder mehr Schwärme mit unterschiedlichen Eigenschaften aufeinandertreffen.

### 3.2.1 Alternierende Stucked Partikel

Ein alternierender Stucked Partikel ist ein eher seltener auftretendes Phänomen. Hierbei handelt es sich um einen Partikel der sich im Stuckedzustand befindet. Wobei sich der Stuckedzustand dieses Partikel darin äußert, dass der Partikel immer zwischen zwei Positionen hin und her springt. Der Partikel alterniert also zwischen diesen beiden Positionen. Diese Art von Stucked Partikeln war der



**Abbildung 3.1:** Der Stucked Partikel alterniert über die Iterationen zwischen den Positionen A und B

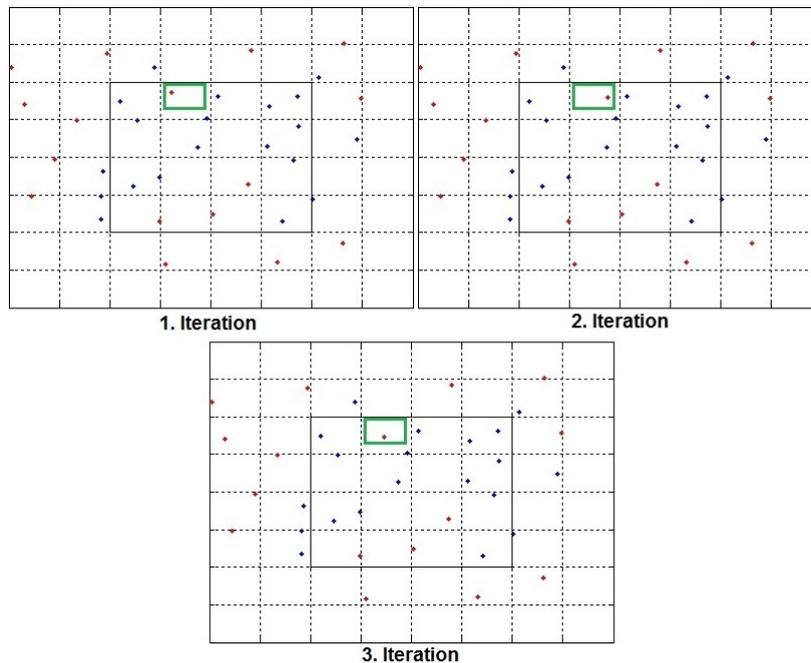
erste naive Ansatz in dieser Arbeit, zur Erklärung des Verhalten eines Stucked Partikels. Im Verlauf der Arbeit wurde jedoch klar das dieses Verhalten, wie oben bereits erwähnt, nur extrem selten auftritt. In Abbildung x.x wird dieses angenommene Phänomen grafisch verdeutlicht.

Der Stucked Partikel befindet sich in der ersten Iteration auf der Position A. Bei der zweiten Iteration springt er auf Position B. Zur dritten Iteration befindet sich der Partikel wieder auf Position A. Im Beispiel aus der Abbildung würde er in der vierten Iteration wieder auf Position B springen, und so weiter. Ausgedehnt auf n Iterationen, würde sich der Partikel bei jeder Ungeraden Iteration auf Position A und bei jeder geraden Iteration auf Position B befinden.

### 3.2.2 Random Move Stucked Partikel

Beim Random Move Stucked Partikel handelt es sich um einen Stucked Partikel, der sich zufällig bewegt. Anders als beim Alternierenden Stucked Partikel, ist die Wahrscheinlichkeit dafür, dass ein Stucked Partikel ein Random Move Stucked Partikel ist sehr hoch. So gut wie jeder Stucked Partikel ist ein Random Move Stucked Partikel.

Ein Random Move Stucked Partikel zeichnet sich dadurch aus, dass er, sobald er seine Stucked Position erreicht hat, in allen folgenden Iterationen einen zufälligen Positionswechsel innerhalb eines kleinen begrenzten Bereiches vollzieht. Dabei nimmt der Partikel allerdings niemals zwei mal hintereinander die gleiche Position ein. In Abbildung x.x ist das Phänomen des Random Move



**Abbildung 3.2:** Der Random Move Stucked Partikel bewegt sich über die Iterationen zufällig innerhalb des **begrenzten Bereiches**

Stucked Partikels illustriert. Es ist gut zu erkennen, dass sich der Partikel über die drei beispielhaft gewählten Iterationen in der Abbildung auf zufällige Positionen innerhalb des grün hervorgehobenen Bereiches bewegt. In allen weiteren Iterationen, die hier nicht dargestellt sind, würde der Partikel sich weiterhin auf andere zufällige Positionen innerhalb dieses Bereiches bewegen. Der Partikel erreicht allerdings niemals eine für ihn optimale Position, die seiner Komfortzone angemessen ist. Der Partikel kann aufgrund seiner Eigenschaften auch niemals aus dem, ihn umgebenden, stabilen Partikelumfeld ausbrechen um nach einer für ihn besseren Position zu suchen. Dies bedeutet der Partikel bewegt sich unendlich weiter und ist aus diesem Grund als Random Move Stucked Partikel zu bezeichnen.

### 3.3 Erkennung von Stucked Partikeln

Um einen Stucked Partikel zu behandeln und so schließlich Stabilität erzeugen zu können, müssen zuerst Möglichkeiten gefunden werden einen Stucked Partikel zu erkennen. Zu diesem Zweck wurden für diese Arbeit eine Reihe von neuen Erkennungsmethoden entwickelt. Diese erheben nicht den Anspruch auf Vollständigkeit, sondern stellen lediglich drei erste Ansätze in Richtung der Stucked Partikel Erkennung dar. Dabei sind sie mal weniger praktikabel, wie im Falle der Erkennung durch Nachbarschafts Veränderungen oder sehr praktikabel wie im Falle der Erkennung anhand von Positionsänderungen eines Partikels und der Erkennung durch Messung der Winkel zwischen den einzelnen Bewe-

gungsvektoren eines Partikels. Alle drei werden im Verlauf dieses Teilabschnittes ausführlich beschrieben.

### 3.3.1 Nachbarschafts Veränderungen

Da in realen Anwendungen, wie zum Beispiel Roboterschwärmen, die Nachbarschaft eines Individuums eine wichtige Rolle spielt, bezieht sich ein Ansatz zur Erkennung einer Stucked Situation auf die Nachbarschafts Veränderungen. Der Grund hierfür liegt darin, dass die Nachbarschaft eines Individuums leicht durch Abfragen der Individuen, die sich innerhalb seines Kommunikationsradius befinden, aufgeschlüsselt werden kann. Bei diesem Ansatz wird von davon ausgegangen, dass sich die Partikel in der Nachbarschaft eines Stucked Partikels, wenn überhaupt, nur geringfügig ändern. Bleiben also über einen bestimmte endliche Anzahl von Zeitschritten die Nachbarn eines Partikels gleich, könnte das ein Hinweis darauf sein, dass dieser Partikel ein Stucked Partikel ist. Bei genauem Nachdenken über diesen Ansatz werden allerdings schnell seine Nachteile deutlich. Auch wenn die Annahme, dass sich die Nachbarschaft eines Stucked Partikels nicht oder nur kaum verändert durchaus korrekt ist, trifft dies nicht auf Stucked Partikel, sondern auch auf eine ganze Menge anderer Partikel zu. Eine Erkennungsmethode über die Nachbarschafts Veränderung würde also einen Stucked Partikel niemals eindeutig herausfiltern können. So würde beispielsweise jeder Partikel, der sich seit mehreren Zeitschritten in einer stabilen Position innerhalb einer ebenfalls stabilen Nachbarschaft befindet, Fälschlicherweise als Stucked Partikel erkannt werden. Ein weiteres Beispiel für eine falsche Detektierung eines Stucked Partikels, wäre ein hypothetischer Partikel, der sich innerhalb einer Gruppe von Partikeln mit relativ stabil bleibenden Abständen zueinander durch den Raum bewegt. Dieser Partikel könnte sich zwar über sehr weite Strecken durch den Raum bewegen, würde aber, aufgrund seiner Nachbarschaftsverhältnisse, nach einiger Zeit Fälschlicherweise als Stucked Partikel identifiziert werden. Aufgrund der vielen mit dieser Methode der Stucked Partikel Erkennung verbundenen Probleme, der vielen möglichen Fälschlicherweise als Stucked definierten Partikel (*false positives*), findet sie in dieser Arbeit keine weitere Betrachtung.

### 3.3.2 Positionsänderung des Partikels

Da ein Stucked Partikel vor Allem dadurch charakterisiert wird, wie er seine Positionen über mehrere Zeitschritte verändert, ist es naheliegend einen Stucked Partikel durch diese Positionsänderungen über eine gewisse Anzahl von Zeitschritten zu detektieren. Diese Methode führt, im Gegensatz zur oben beschriebenen Erkennung durch Nachbarschafts Veränderungen, zu präzisen Lokalisierungen von Stucked Partikeln. Der Grund dafür liegt darin, dass bei Betrachtung der Positionsänderungen eines Partikels spezielle Eigenschaften, welche einen Stucked Partikel eindeutig ausmachen und von allen anderen Partikeln abheben, berücksichtigt werden können. Dabei wurden zwei Mögliche Herangehensweisen zur Erkennung von Stucked Partikeln entwickelt. Die Erkennung alternierender Stucked Partikel, dieser stellt ein eher naiven und frühen Ansatz dar, welcher ausschließlich zur Erfassung Alternierender Stucked Partikel dient. Er ist damit nicht sehr praktikabel, da alternierende Stucked Partikel, wie bereits erwähnt, nur eine sehr kleine und seltene Teilmenge von Stucked Partikeln

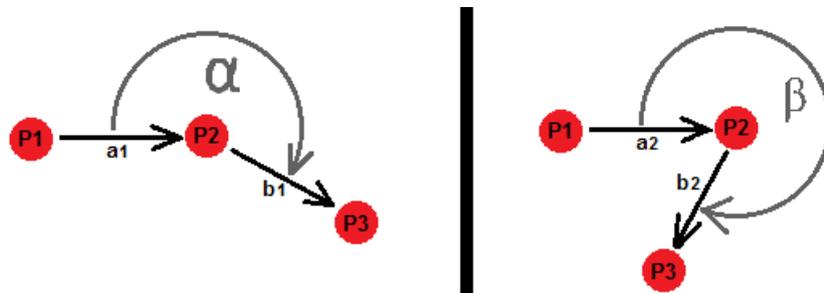
ausmachen. Der zweite Ansatz ist die Erkennung von Random Move Stucked Partikeln. Dieser dient der Erfassung jeglicher Stucked Partikel, einschließlich alternierender Stucked Partikel. Er ist damit der allgemeinere und wesentlich bessere Ansatz. Im Folgenden werden beide Ansätze ausführlich erläutert:

- **Erkennung von alternierenden Stucked Partikeln:**

Einen alternierenden Stucked Partikel, anhand seiner Positionsänderungen über mehrere Zeitschritte zuerkennen, ist trivial. Zu aller Erst muss man sich zu diesem Zweck die charakteristischen Bewegungsmuster vor Augen führen, welche einen alternierenden Stucked Partikel ausmachen. Ein solcher Stucked Partikel springt, sobald er seinen Stuckedzustand erreicht hat, nur noch zwischen zwei festen Positionen hin und her. In diesem Fakt liegt bereits die Lösung zur Detektion eines solchen alternierenden Stucked Partikels. Wenn jeder Partikel die Positionen abspeichert an denen er bereits gewesen ist, sind nur noch wenige simple Vergleiche nötig um einen alternierenden Stucked Partikel eindeutig zuerkennen. So muss die Aktuelle Position des Partikels ungleich seiner vorherigen Position sein und die aktuelle Position muss identisch zur Position zwei Zeitschritte zuvor sein. Des Weiteren muss vorherige Position des Partikels identisch zu seiner Position vor drei Zeitschritten sein. Dabei ist offensichtlich, dass man durch diese Vorgehensweise ausschließlich alternierende Stucked Partikel erkennen kann, da nur deren spezielle Eigenschaft in den Vergleichen Berücksichtigung finden. Die Vergleiche lassen sich allerdings sehr einfach noch beliebig auf die Positionen vor vier beziehungsweise vor fünf Zeitschritten, und so weiter ,ausdehnen. Dadurch kann eine noch präzisere Aussage über den Zustand des Partikels getroffen werden. Hierbei sollte man sich jedoch darüber im Klaren sein, dass mehr Vergleiche zwar höhere Genauigkeit, aber in den meisten Fällen auch schlechtere Performance in Bezug auf die Berechnungsgeschwindigkeit bedeuten. Wenn man davon ausgeht, dass die beschriebenen Vergleiche bei einer Computersimulation unter Umständen für eine sehr hohe Anzahl ( $> 1000$ ) von Partikeln durchgeführt werden müssen, ist dies ein auch für heutige Rechensysteme nicht unerhebliches Problem. Dieses wird noch signifikanter bei der aufwändigeren, aber allgemeineren Erkennung von Random Move Stucked Partikeln, welche im Folgenden aufgeschlüsselt wird.

- **Erkennung von Random Move Stucked Partikeln:**

Der Erkennung von Random Move Stucked Partikeln bietet, im Gegensatz zur Erkennung alternierender Stucked Partikel, den Vorteil, dass sie die Detektion jeglicher Stucked Partikel ermöglicht. Dies schließt selbstverständlich alternierende Stucked Partikel ein, da diese nur eine Teilmenge der Random Move Stucked Partikel darstellen. Zum Erkennen von Random Move Stucked Partikeln müssen eine Menge von Bedingungen untersucht werden. So ist Entfernung zwischen den Positionen, welche ein Stucked Partikel einnimmt nicht sehr groß, da der Partikel eine zufällige Bewegung in einem relativ kleinen Gebiet ausführt. Des Weiteren bewegt sich solch ein Partikel bei einer Iteration in positive Richtung und bei der nächsten Iteration in negative Richtung. Diese Verhaltensweise sollte sowohl für die



**Abbildung 3.3:** Die Winkel zwischen den Bewegungsvektoren zwei verschiedener Partikel. Zwischen den Vektoren auf linken Seite liegt ein kleinerer Winkel vor, als zwischen denen auf der rechten Seite

x- als auch für die y-Richtung untersucht werden. Wenn also ein solches Verhalten über mehrere Iterationen in x- oder y-Richtung, einhergehend mit relativ kleiner Positionsänderungsgröße, erfasst wurde, liegt ein Random Move Stucked Partikel vor. Um eine fundierte Aussage darüber treffen zu können ob ein Partikel wirklich ein Random Move Stucked Partikel ist oder nicht, darf deswegen nicht nur die aktuelle Position mit der Position der vorangegangenen Iteration verglichen werden. Um solch einen Vergleich korrekt für einen Partikel durchführen zu können, müssen also die Positionen die er bereits eingenommen hat, bis zu einem gewissen Grad oder am besten alle, gespeichert werden. Da diese Speicherung der Positionen für jeden Partikel ein Simulation zu jeder Iteration geschehen muss und eine Simulation unter Umständen über eine große Anzahl von Iterationen (> 1000) läuft, kann diese Art der Stucked Partikel Erkennung anhand der Positionsänderungen, sehr aufwändig werden.

### 3.3.3 Winkelmessung zwischen Bewegungsvektoren

Eine weitere Möglichkeit einen Stucked Partikel zu erkennen, besteht im Abmessen der Winkel zwischen der aktuellen Position des Partikel und seiner beiden vorangegangenen Positionen. Also in der Nachverfolgung der Bewegungsvektoren, welche der Partikel von seiner vorletzten zu seiner letzten Position hatte und von seiner letzten zu seiner aktuellen Position hat. Der Winkel zwischen diesen Bewegungsvektoren des Partikels kann Aufschluss darüber geben ob der Partikel stucked ist oder nicht. Zu diesem Zweck werden zwei Vektoren aus den Positionen eines Partikels bestimmt. Der erste Vektor aus vorletzter und letzter Position des Partikels und der zweite Vektor aus letzter und aktueller Position des Partikels. Ist der Winkel zwischen den beiden bestimmten Vektoren größer als  $270^\circ$ , dann liegt ein Wechsel der Bewegungsrichtung des Partikels von einer Richtung des Raumes in die Andere vor. Dieses Verhalten kann auf einen Stucked Partikel hindeuten. Hierbei soll angemerkt seine, dass diese Erkennungsmethode auch vice versa in die Andere Richtung mit Winkeln kleiner als  $90^\circ$  funktioniert. Sie wird hier der Einfachheit halber allerdings nur am Beispiel der großen Winkel veranschaulicht. Abbildung 3.3 verdeutlicht den Sachverhalt. In der linken Darstellung vollführt der Partikel eine Bewegung in positive x-Richtung. Der Winkel  $\alpha$  zwischen Vektor  $a_1$  und Vektor  $b_1$  ist klei-

ner als  $270^\circ$ . Dadurch würde der Partikel in der linken Darstellung nicht als Stucked Partikel detektiert werden. In der rechten Darstellung hingegen bewegt sich der Partikel zuerst in positive x-Richtung, aber danach in plötzlich in negative x-Richtung. Der Winkel  $\beta$  zwischen den Vektoren  $a_2$  und  $b_2$  ist dadurch größer als  $270^\circ$ . Dies lässt darauf schließen, dass der Partikel in der rechten Darstellung ein Stucked Partikel sein könnte. Um eine genauere Aussage über den Zustand des Partikels treffen zu können, kann die Erkennungsmethode verfeinert werden. Dies kann geschehen indem der Abstand zwischen den Positionen berechnet wird. Unterschreitet der Abstand der Positionen einen bestimmten Wert, ist die Wahrscheinlichkeit höher, dass ein Stucked Partikel vorliegt. Der Wert sollte je nach Simulation oder Anwendungsgebiet individuell angepasst werden. Um die Aussage über noch präziser zu gestalten, könnte man zusätzlich nicht nur den Winkel zwischen den zwei letzten Bewegungsvektoren eines Partikels messen, sondern auch den zwischen denen vorhergehenden. Wenn vier der letzten fünf Winkel größer als  $270^\circ$  sind, kann man mit hoher Sicherheit sagen, dass ein Stucked Partikel vorliegt. Bei diesem Vorgehen sollte beachtet werden, dass die Berechnung von mehreren Vektoren, die Abmessung vieler Winkel und die Abstandsberechnungen ein höherer Rechenaufwand zu erwarten ist. Dadurch wird dieser Erkennungsmethode zwar genauer, benötigt aber auch mehr Performance und eine längere Laufzeit. Die Anzahl der zu messenden Winkel entspricht immer der Anzahl der berechneten Vektoren verringert um eins, oder der Anzahl der berechneten Abstände verringert um eins. ( $Anzahl\ Winkel = Anzahl\ Vektoren - 1 \parallel Anzahl\ Abstände - 1$ ) Wurde ein Stucked Partikel schließlich als solcher erkannt, kann seine Behandlung erfolgen. Dies ist Thema des nächsten Abschnitts.

### 3.4 Behandlung von Stucked Partikeln

Für die Praxis ist es sehr wichtig stabile Schwärme erzeugen zu können. Das bedeutet Schwärme bei denen die Individuen an statisch an einer festen Position im Raum verharren. Nur so kann es später ermöglicht werden Schwärme in Gänze zu kontrollieren und so für zukünftige realweltliche Anwendungen nutzen zu können. Das Phänomen der Stucked Partikel sorgt allerdings beim Wechselwirken mehrerer Schwärme für Instabilität. Aus diesem Grund ist unumgänglich Wege zu finden Stucked Partikeln entgegenzuwirken. Für diese Arbeit wurden zwei mögliche Vorgehensweisen für das Behandeln von Stucked Partikeln entwickelt. Zu Einem die triviale Variante der Behandlung durch durch simples Festhalten. Zum Anderen die wesentlich komplexere und vielversprechende Variante der Anpassung von Partikeln im Stuckedzustand durch adaptive Radien. Beide Methoden werden in den nachfolgenden Teilabschnitten ausführlich veranschaulicht.

#### 3.4.1 Festhalten

Während des Entstehungsprozesses dieser Arbeit wurde im Bezug auf das Problem der Behandlung von Stucked Partikel zu aller Erst die Möglichkeit erwogen, Partikel die sich im Stuckedzustand befinden mit einem simplen Stoppen ihrer Bewegung entgegenzuwirken. Bei dieser, treffend als Festhalten bezeichneten, Methode wird ein Partikel, nachdem er als stucked detektiert wurde, einfach

an seiner momentanen Position im Stuckedbereich zum Stillstand gebracht. Die Bewegung des Stucked Partikels wird also gestoppt und sein Stuckedzustand so aufgelöst. Dadurch wird der Partikel künstlich von einem Stucked Partikel zu einem stabilen Partikel gemacht. Diese Vorgehensweise ist simpel, schnell und wirkungsvoll, sorgt aber nicht für optimale Ergebnisse. Aus diesem Grund würde für diese Arbeit eine weitere Methode entwickelt, welche vielseitiger Stucked Partikel behandelt kann und so für Ergebnisse sorgen kann die optimaler sind. Diese Methode wird im folgenden Teilabschnitt im Detail aufgeschlüsselt.

### 3.4.2 Adaptive Radien

Da die Behandlungsmethode mit simplen Festhalten der Partikel erschien im Verlauf der Erstellung der Arbeit als zu starr. So wurde das Konzept der Stucked Partikel Behandlung mit Hilfe der adaptiven Anpassung der Repulsion- und Attractionradien von Stucked Partikeln entwickelt. Dabei werden der Repulsionradius und der Attractionradius eines Partikels, welcher als Stucked erkannt wurde, so lange leicht verringert, bis der Stucked Zustand des Partikel aufgelöst ist. Sollte ein Radius hierbei den Wert 0 erreichen, wird dieser wieder auf seinen Ausgangswert zurückgesetzt und dann wieder neu verringert, bis er Stabilität erreicht hat. Dies wird getan, weil jeder Partikel, welcher Nullradien erreicht hat, stehenbleiben würde. Ein stehender Partikel erfüllt zwar das Ziel stabil zu sein. Solch ein Partikel würde sich aber in diesem Fall nicht mehr von einem stabilen Partikel, welche durch das Vorgehen mit einfachen festhalten erreicht wurde, unterscheiden. Da das Vorgehen mit adaptiven Radien allerdings möglichst optimale Lösungen hervorbringen soll, wird mit dem Zurücksetzen auf den Ursprungswert gearbeitet. Ein Nullradius ist niemals eine optimale Lösung, weil das Problem der Instabilität nur aufgelöst wird, indem man dem Stucked Partikel das Schwärmverhalten nimmt. Alle Rechnungen, welche das Schwärmverhalten auslösen nehmen, durch die Nullradien, automatisch den Wert null an und nicht dadurch, dass eine optimale Wertekonstellation (*optimale Position*) gefunden wurde. Bei der Radienanpassung gibt es zwei verschiedene Vorgehensweisen. Diese sind die etwas starrere Anpassung mit einen festen Wert und die dynamischere prozentuale Anpassung. Außerdem wurde für die Radienanpassung noch eine Erweiterungsmöglichkeit entwickelt. Bei dieser Erweiterung, werden bei Stucked Partikel zusätzlich zur Radienverringerng noch ihre Bewegungsgeschwindigkeit erhöht. Alle drei Ansätze werden im folgenden Verlauf dieses Teilabschnittes detailliert beschrieben:

- **Radienanpassung mit festen Werten:**

Die starrste Methode der adaptiven Radienanpassung, ist die Anpassung mit Hilfe eines fest definierten Wertes. Hierbei wird ein immer gleichbleibender Wert zur Verkleinerung der Radien im Stuckedfall verwendet. Diese Anpassung kann sehr grob erfolgen, indem man einen sehr großen Anpassungswert wählt oder man kann sie feiner gestalten indem man einen sehr kleinen Wert wählt. Dabei sollte beachtet werden, dass ein sehr kleiner Wert zwar für eine bessere Anpassung sorgen kann, jedoch unter Umständen mit sehr hohem Rechenaufwand verbunden ist. Mathematisch gesehen wird der feste Wert bei jeder Iteration, in der ein Partikel als stucked erkannt ist, von den Radien dieses Partikels subtrahiert. Hierbei

muss darauf geachtet werden, dass niemals ein negativer Radienwert entstehen kann. Aus diesem Grunde entsprechende Abfragen implementiert werden, welche dies verhindern.

- **Prozentuale Radienanpassung:**

Eine dynamischere Methode als die Radienanpassung mit Hilfe fester Werte, stellt die Prozentuale Radienanpassung dar. Hierbei wird aus dem aktuellen Wert des Radius ein neuer Radius für die folgende Iteration berechnet, welche einem festgelegtem Prozentsatz des aktuellen Radius entspricht. Es wird klar, dass sich die Radien dadurch bei jeder Iteration um einen etwas kleineren Wert verringern. Dies sorgt für eine höhere Genauigkeit, je kleiner die Radien werden. Eine Eigenschaft, welche die prozentuale Radienanpassung genauer und in manchen Fällen auch schneller als die Radienanpassung mit festen Werten macht. Mathematisch gesehen werden bei dieser Methode der Radienanpassung der aktuellen Radien mit einem prozentualen Wert multipliziert. Dieser Wert entspricht dem Prozentsatz der aktuellen Radien, welche die Radien in der nächsten Iteration annehmen sollen. Möchte man also, dass die Radien in der nächsten Iteration 90% der aktuellen Radiengröße besitzen, würde man die aktuellen Werte der Radien mit 0,90 multiplizieren.

- **Beschleunigung bei Radienanpassung:**

Um Stucked Partikel zu behandeln, kann man zusätzlich zum adaptiven Anpassen ihrer Radien, die Geschwindigkeit erhöhen, mit der sie sich durch den Raum bewegen. Um solche eine Beschleunigung im Stuckedfall zu erreichen, wird zusätzlich zu Verringerung des Repulsion- und Attractionsradius eines Partikel auch sein Positionsänderungswert in x- sowie in y-Richtung um einen bestimmten festen oder prozentualen Wert erhöht. Hier muss allerdings darauf geachtet werden, dass kein zu großer Wert zur Erhöhung genutzt wird, da der Partikel sonst möglicherweise aus dem Ausdehnungsgebiet der Schwärme und damit aus dem Einflussbereich aller anderen Partikel springen könnte. Die Idee für diese Erweiterung entstand aus der Annahme, dass sich ein Stucked Partikel möglicherweise noch schneller aus seiner Stuckedsituation befreien kann, wenn er sich mit erhöhter Geschwindigkeit bewegt.

# Kapitel 4

## Implementierung

Die grundlegende technische Realisierung des Simulationstools zur Simulation mehrerer Partikelschwärme, ist Hauptbestandteil dieses Kapitels. Zu aller Erst wird auf die, zur technischen Umsetzung genutzten Hilfsmittel und Werkzeuge Bezug genommen. Im Anschluss wird die programmiertechnische Umsetzung der entwickelten Ansätze detailliert beschrieben, dabei werden auch *Pseudocode*<sup>1</sup> Darstellungen zur besseren Veranschaulichung genutzt. Da für das Simulationstool, zur besseren Übersicht und schnelleren Bedienung, eine eigene GUI entwickelt wurde, wird auch auf die Gestaltung dieser GUI im letzten Abschnitt dieses Kapitels eingegangen.

### 4.1 Verwendete Werkzeuge und Hilfsmittel

Die gesamte Arbeit wurde ausschließlich mit Zuhilfenahme der Programmierumgebung MatLab, in der Version R2012b, implementiert. Hierbei wurde zur Entwicklung der grafischen Benutzeroberfläche das ebenfalls in MatLab integrierte GUIDE Werkzeug genutzt. Dieses dient explizit der einfacheren Erstellung von grafischen Benutzeroberflächen mit MatLab und bietet ein Framework in dem einfache und gängige GUI-Elemente, wie zum Beispiel Buttons, Edit Text Boxen oder Silder , bereits grundlegend vorimplementiert sind. Eine weiter sehr große Erleichterung stellte die MatLab Dokumentation dar. In dieser sind detaillierte Information zu fast allen in der MatLab-Bibliothek vorimplementierten Klassen und Funktionen aufgeführt. Diese Vorimplementierungen in der MatLab-Bibliothek boten eine große Aufwandsreduzierung, vor Allem im Bezug auf die Implementierung mathematischer Werkzeuge, und finden aus diesem Grund in der technischen Umsetzung der Arbeit an einigen Stellen Verwendung.

### 4.2 Programmiertechnische Umsetzung

Die für die Simulationen in dieser Arbeit genutzten Methoden zur Realisierung des Schwärmverhaltens basieren vollständig auf den bereits in [Blum, 2013] umgesetzten Code. Der unterschied besteht darin, dass diese Methoden hier

---

<sup>1</sup>Pseudocode: Bezeichnet Programmcode, welcher nicht der direkten programmiertechnischen Umsetzung dient, sondern lediglich der besseren Veranschaulichung bestimmter Sachverhalte

zu Simulation mehrerer Schwärme, anstelle nur eines Schwarmes, genutzt werden. Das grundlegende Prinzip der technischen Umsetzung des Schwärmverhaltens ist in Pseudocode 1 veranschaulicht. Dieser Pseudocode stellt nur eine

---

**Pseudocode 1** Berechnung des Schwärmverhaltens

---

**Input:** Anzahl der Partikel im Schwarm ( $numPart$ )

Anzahl der Iterationen ( $numIt$ )

Größe des Repulsionradius ( $rr$ )

Größe des Attractionradius ( $ar$ )

alte Positionen ( $oldP$ )

**Output:** neue Positionen ( $newP$ )

```

1: for  $i = 1$  to  $i == numIt$  do
2:   for  $j = 1$  to  $j == numPart$  do
3:     berechne Distanz zwischen den Partikeln
4:      $numParrr \leftarrow$  Anzahl Partikel in  $rr$  von  $j$ 
5:      $numPartra \leftarrow$  Anzahl Partikel in  $ra$  von  $j$ 
6:     if ( $numParrr > 0$ )
7:       then
8:          $PosCh \leftarrow$  Berechne Positionsänderung mit Gleichung für Abstoßung
9:       end if
10:    if ( $numPartra > 0$ )
11:      then
12:         $PosCh \leftarrow$  Berechne Positionsänderung mit Gleichung für Anziehung
13:      end if
14:     $newP(j) \leftarrow oldP(j) + PosCh$ 
15:   end for
16: end for
17: return  $newP$ 

```

---

grobe Veranschaulichung. Der Algorithmus nimmt die Anzahl der Partikel in einem Schwarm, die Anzahl der Iterationen, den Repulsionradius, sowie den Attractionradius und die alten Partikelpositionen als Eingabewerte und berechnet daraus die neuen Positionen der Partikel. Dies wird für jede Iteration jeweils für alle Partikel getan. Eine detaillierte Beschreibung dieses Algorithmus ist in [Blum, 2013] zu finden, dort ist noch zusätzlich ein Turbulenzfaktor im Algorithmus vorhanden auf den hier verzichtet wurde, da dieser für die hier durchgeführten Untersuchungen unerheblich ist. Durch den Algorithmus kann das Schwärmverhalten der Partikel simuliert werden. Er läuft in den Simulationen der Arbeit lediglich über mehrere, anstatt über einen einzigen Schwarm. Der Repulsion- und der Attractionradius wird verändert, je nachdem zu welchem Schwarm der momentan betrachtete Partikel gehört. Bei der programmier-technischen Umsetzung liegt das Hauptaugenmerk dieser Arbeit nicht auf dem Schwärmverhalten, weshalb dieses hier auch nicht detaillierter betrachtet wird. Es liegt vor allem in der technischen Umsetzung der Erkennung und Behandlung von Stucked Partikeln.

Zur Erkennung jeglicher Arten von Stucked Partikeln werden in der Implementierung nur simple If-Abfragen genutzt. Die Behandlung erfolgt dann, wenn die Abfragen zutreffen, was ausschließlich der Fall ist, wenn ein Stucked Partikel vorliegt. Alle Erkennungsarten können mit allen Behandlungsarten kombiniert

werden. Im Pseudocode 2 ist der Algorithmus zur Erkennung von alternierenden Stucked Partikeln durch ihre Positionsänderung aufgeführt. Da Stucked Partikel

---

**Pseudocode 2** Erkennung von alternierenden Stucked Partikeln durch Positionsänderungen

---

**Input:** Zu untersuchender Partikel (Part)  
 Zahl der aktuellen Iteration (aktI)  
**Output:** Zustand des Partikels (isSt)  
 if ((Part.position(aktI) == Part.position(aktI-2))  
 && (Part.position(aktI-1) == Part.position(aktI-3)))  
 then  
 2: *isSt* ← *true*  
 else  
 4: *isSt* ← *false*  
 end if  
 6: **return** isSt

---

sehr selten sind, ist dieser Algorithmus eher unpraktikabel. Er soll aber trotz alledem hier nicht unerwähnt bleiben, da eine wichtige Rolle bei der Entstehung dieser Arbeit spielte. Der Algorithmus nimmt den Partikel, dessen Zustand untersucht werden soll und zusätzlich die Zahl der aktuellen Iteration. Daraus erkennt er den Zustand des Partikels. Das bedeutet der Algorithmus erkennt ob der Partikel ein alternierender Stucked Partikel ist oder nicht und gibt diese Information aus. Dabei wird mit einer einfachen If-Abfrage gearbeitet. Es wird ausschließlich getestet ob der Partikel in der vorletzten Iteration die gleiche Position hatte, wie in der aktuellen Iteration und ob der Partikel in der letzten Iteration die gleiche Position hatte, wie drei Iterationen vor der aktuellen. Trifft dies zu, dann ist der Partikel ein alternierender Stucked Partikel, ansonsten nicht.

Der Algorithmus zur Erkennung von Random Move Stucked Partikeln ist ein wesentlich praktikablen Ansatz, als der grade beschriebene. Er ist in Pseudocode 3 dargestellt. Dieser Algorithmus nimmt den zu untersuchenden Partikel, die Zahl der aktuellen Iteration und den maximal erwünschten Abstand zwischen aufeinanderfolgenden Positionen als Eingabewerte. Als Ausgabewert gibt er Auskunft darüber, ob der untersuchte Partikel ein Stucked Partikel ist oder nicht. Im Algorithmus wird der Test ob ein Stucked Partikel vorliegt durch If-Abfragen realisiert. Dabei wird sowohl in x-Richtung, also auch in y-Richtung geprüft, ob der Partikel, in den letzten drei Iterationen, eine Bewegungsrichtungsänderung vollführt hat. Dies geschieht indem abgefragt wird, ob die aktuelle Position des Partikel kleiner (*beziehungsweise größer*) ist als die letzte Position und ob die letzte Position dann wiederum größer (*beziehungsweise kleiner*) ist als die vorletzte Position. Trifft ein dieser Abfragen für die x- oder y-Richtung zu, wird noch der Abstand überprüft. Ist der Abstand zwischen aktueller und letzter Position und zwischen letzter und vorletzter Position kleiner oder höchstens gleich des Maximalabstands, wird der Partikel als stucked bewertet. Ansonsten wird Partikel standardmäßig als nicht stucked bewertet. Um die Bewertung genauer zu gestalten, könnte man optional die Abfrage nach der Bewegungsrichtungsänderung noch weiter zurück ausdehnen.

Eine ähnliche Herangehensweise wird bei der Stucked Partikel Erkennung

---

**Pseudocode 3** Erkennung von Random Move Stucked Partikeln durch Positionsänderungen
 

---

**Input:** Zu untersuchender Partikel (Part)  
 Zahl der aktuellen Iteration (aktI)  
 maximaler Abstand zwischen den Positionen (maxDis)

**Output:** Zustand des Partikels (isSt)

```

isSt ← false
if ((Part.position(aktI).x < Part.position(aktI-1).x)
  && (Part.position(aktI-1).x > Part.position(aktI-2).x))
  || (Part.position(aktI).x > Part.position(aktI-1).x)
  && (Part.position(aktI-1).x < Part.position(aktI-2).x))
  || (Part.position(aktI).y < Part.position(aktI-1).y)
  && (Part.position(aktI-1).y > Part.position(aktI-2).y)
  || (Part.position(aktI).y > Part.position(aktI-1).y)
  && (Part.position(aktI-1).y < Part.position(aktI-2).y))
  then
3:   if (Abstand(Part.position(aktI), Part.position(aktI-1)) <= maxDis &&
        Abstand(Part.position(aktI-1), Part.position(aktI-2)) <= maxDis ) then
        isSt ← true
        end if
6: end if
return isSt
  
```

---

durch Winkelmessung genutzt. Hier wird ebenfalls auf eine Bewegungsrichtungsänderung geprüft. Dies geschieht jedoch auf andere Art und Weise als beim vorherigen Algorithmus. Die Vorgehensweise ist in Pseudocode 4 veranschaulicht. Der Algorithmus zur Erkennung mit Hilfe von Winkelmessung nimmt als Eingabewerte den zu untersuchenden Partikel, die Zahl der aktuellen Iteration und die Anzahl wie viel Iterationen die Abfrage zurückreichen soll. Durch den letzten Eingabewert kann die Genauigkeit Erkennung festgelegt werden. Es beachtet werden, dass dieser nicht kleiner als zwei sein drauf, da sonst der Algorithmus nicht korrekt arbeiten kann. Der Ausgabewert des Algorithmus ist, der Zustand des Partikels. Im Algorithmus wird zuerst ein Array erstellt, welches Vektoren abspeichern kann. Im Anschluss wird eine Variable initialisiert, die dazu dient alle Winkel, welche größer als  $270^\circ$  zu zählen. Das Array wird innerhalb einer For-Schleife mit Vektoren gefüllt. Innerhalb der Schleife, werden die Vektoren zwischen den Positionen berechnet. Dies sind die Bewegungsrichtungsvektoren. Die Berechnung geschieht bis zur gewünschten Betrachtungstiefe. Nach Durchlaufen der Schleife, ist das Array mit Bewegungsrichtungsvektoren gefüllt. Es folgt eine weitere For-Schleife, welche das Array durchläuft. Dabei wird die Größe des Winkels zwischen zwei, im Array (*und in Wirklichkeit*), nebeneinanderliegenden Bewegungsvektoren berechnet. Ist der Winkel größer als  $270^\circ$ , wird die Winkelzählvariable um eins erhöht. Nachdem die Schleife durchlaufen ist, folgt eine letzte If-Abfrage. In dieser Abfrage wird untersucht, ob die Anzahl der Winkel größer als  $270^\circ$  höchstens um eins kleiner ist, als die Anzahl der vorhandenen Winkel. Die Anzahl der vorhandenen Winkel entspricht immer genau der Anzahl an vorhandenen Bewegungsvektoren verringert um eins ( $VektArr.length - 1$ ). Aus diesem Grund wird getestet ob

---

**Pseudocode 4** Erkennung von Stucked Partikeln durch Winkelmessung

---

**Input:** Zu untersuchender Partikel (Part)  
Zahl der aktuellen Iteration (aktI)  
Anzahl wie viele Iterationen zurück betrachtet werden sollen (z) [Mindestwert 2]

**Output:** Zustand des Partikels (isSt)  
*VektArr*  $\leftarrow$  Array zum Abspeichern der Vektoren  
*numAng*  $\leftarrow$  Anzahl der Winkel größer  $270^\circ$  [am Beginn 0]

```

for  $i = 1$  to  $i = z$  do
4:   VektArr( $i - 1$ )  $\leftarrow$  berechne Vektor zwischen Part.position((aktI- $z$ )+( $i-1$ ))
      und Part.position((aktI- $z$ )+ $i$ )
end for
for  $j = 0$  to  $j =$  VektArr.length - 1 do
   Angle  $\leftarrow$  Messe Winkel zwischen VektArr( $j$ ) und VektArr( $j+1$ )
8:   if (Angle >  $270^\circ$ )
     then
       numAng  $\leftarrow$  numAng + 1
     end if
end for
12: if (numAng >= (VektArr.length - 2) && (VektArr.length > 2)
    || numAng == 1 && (VektArr.length == 2)) then
   isSt  $\leftarrow$  true
else
   isSt  $\leftarrow$  false
16: end if
return isSt
```

---

die Anzahl der Winkel größer  $270^\circ$  ( $numAng$ ) größer oder gleich der Anzahl der Bewegungsvektoren verringert um zwei ist ( $VektArr.length - 2$ ). Außerdem wird der Sonderfall, nur eines vorhandenen Winkels, abgefangen. Hierbei muss dieser eine Winkel größer als  $270^\circ$  sein. Ohne diese Zusatzbedingung würde, im Falle nur eines vorhandenen Winkels, jeder Partikel als stucked gewertet werden.

Würde ein Partikel als stucked erkannt, kann dieser behandelt werden, um seinen stucked Zustand aufzulösen. Der Algorithmus, welcher dies tut, ist in Pseudocode 5 aufgelistet. Dieser Algorithmus hat eine große Menge an Eingabewerten. Dies sind die zu verwendende Behandlungsart, der Repulsionradius des Partikels, der Attractionradius des Partikels, der Wert zur Anpassung mit festem Wert und der prozentuale Wert für die prozentuale Anpassung. Die Ausgabewerte des Algorithmus sind der neue Repulsionradius und der neue Attractionradius des behandelten Partikels. Als erstes weist der Algorithmus dem neuen Repulsion- und Attractionradius die Werte ursprünglichen Radien des Partikels zu. Anschließend beginnt eine While-Schleife, welche solange durchlaufen wird, bis sich der behandelte Partikel nicht mehr im stucked Zustand befindet. Für den Test ob der Partikel stucked ist, können die Algorithmen aus Pseudocode 2 oder 3 genutzt werden. Innerhalb der While-Schleife befinden sich If-Abfragen durch die festgestellt wird, welche der drei Behandlungsmethoden genutzt werden soll. Wenn die Methode durch Festhalten verwendet werden soll, wird den neuen Radien lediglich der Wert null zugewiesen. Dadurch bewegt sich der Partikel nicht mehr und ist somit nicht mehr im Stuckedzustand.

Wenn die Radienadaptation mit Hilfe eines festen Wertes zur Behandlung genutzt werden soll, wird das Vorgehen komplexer. Hier wird Abgefragt, ob die neuen Radien noch größer als der feste Wert zur Verringerung sind. Dies geschieht zum Einen um zu verhindern, dass die neuen Radien kleiner oder gleich null werden. Zum Anderen um festzustellen, ob die neuen Radien auf die ursprünglichen Radienwerte zurückgesetzt werden müssen, da sie zu klein sind. Sind die neuen Radien nicht zu klein, wird der feste Wert von ihnen subtrahiert.

Falls eine Behandlung durch prozentuale Radienanpassung durchgeführt werden soll, sind die Abfragen ähnlich zur Behandlung durch einen festen Wert. Hier wird abgefragt, ob die neuen Radien einen Wert von 0,1 nicht unterschreiten. Der Wert 0,1 wird in dieser Arbeit benutzt, muss aber nicht zwingend verwendet werden. Hier kann ein beliebiger positiver Wert eingefügt werden, welcher nicht unterschritten werden darf. Dieser hängt davon ab, wie klein man die Radien werden lassen möchte. Ist der Wert nicht unterschritten, werden die neuen Radien durch einen prozentualen Wert ihrer Selbst ersetzt. Wird der Wert unterschritten werden die neuen Radien auf die ursprünglichen Radienwerte zurückgesetzt.

Sobald die Bedingung für die While-Schleife nicht mehr erfüllt ist, werden die neuen Werte der Radien ausgegeben.

Um die Stabilisierung zu beschleunigen, kann die Stucked Partikel Behandlung um eine Komponente erweitert werden. Diese Erweiterung ist die Beschleunigung der Bewegung eines Partikels im Stuckedfall. Der dafür genutzte Algorithmus ist im Pseudocode 6 dargestellt. Die Eingabewerte des Algorithmus sind die aktuelle Position des Partikels, die Information darüber ob die Beschleunigungserweiterung genutzt werden soll und der Wert um den beschleunigt werden soll. Dieser Beschleunigungswert muss positiv sein, da sich sonst die Position des Partikels invertiert. Der Wert könnte außerdem zum Verlangsamen der Partikelbewegung eingesetzt werden, indem man einen Wert kleiner als eins für ihn wählt.

---

**Pseudocode 5** Behandlung von Stucked Partikeln
 

---

**Input:** Behandlungsart (Bart)  
 Größe des Repulsionradius des Partikels (rr)  
 Größe des Attractionradius des Partikels (ar)  
 Wert für feste Radienadaption (value)  
 Prozentwert für prozentuale Radienadaption (procent)

**Output:** neuer Repulsioradius (newrr)  
 neuer Attractionradius (newar)  
 $newrr \leftarrow rr$   
 $newar \leftarrow ar$

**while** (*Part.isSt == true*) **do**  
   **if** (*Bart == Festhalten*)  
     **then**  
 5:      $newrr \leftarrow 0$   
        $newar \leftarrow 0$   
     **end if**  
   **if** (*Bart == Adaptiv mit festen Wert*)  
     **then**  
       **if** ( $newrr > value$ ) **then**  
 10:      $newrr \leftarrow newrr - value$   
       **else**  
          $newrr \leftarrow rr$   
       **end if**  
       **if** ( $newar > value$ ) **then**  
 15:      $newar \leftarrow newar - value$   
       **else**  
          $newrr \leftarrow ar$   
       **end if**  
     **end if**  
 20:  **if** (*Bart == Adaptiv prozentual*)  
     **then**  
       **if** ( $newrr \geq 0,1$ ) **then**  
          $newrr \leftarrow newrr * procent$   
       **else**  
          $newrr \leftarrow rr$   
 25:     **end if**  
       **if** ( $newar \geq 0,1$ ) **then**  
          $newar \leftarrow newar * procent$   
       **else**  
          $newrr \leftarrow ar$   
 30:     **end if**  
     **end if**  
   **end while**  
**return** newrr, newar

---

---

**Pseudocode 6** Erweiterung zur Beschleunigung von Stucked Partikeln

---

**Input:** aktuelle Position des Partikels (*Pos*)  
Beschleunigungserweiterung eingeschaltet (*ext*)  
Beschleunigungswert (*acc*) [muss positiv sein]  
**Output:** neue Position des Partikels (*newPos*)  
*newPos*  $\leftarrow$  *berechne neue Position aus Pos* [mit Hilfe von Pseudocode 1]  
**if** (*ext* == *true*) **then**  
    *newPos*  $\leftarrow$  *newPos*\**acc*  
**end if**  
**return** *newPos*

---

Dies wird in dieser Arbeit nicht getan. Als Ausgabewert gibt der Algorithmus die neue Position des Partikels zurück. Im Algorithmus wird zuerst der Wert für die neue Position aus der alten Position berechnet. Die Berechnung erfolgt mit Hilfe des Algorithmus aus Pseudocode 1. Wenn die Beschleunigungserweiterung genutzt werden soll, wird der Wert der neuen Position mit dem Beschleunigungswert multipliziert. Schließlich gibt der Algorithmus den Wert der neuen Position aus.

Die Implementierung beschränkt sich nicht nur auf die grade beschriebenen Algorithmen. Zusätzlich wurde eine grafische Benutzeroberfläche erstellt. Auf deren Aufbau und Funktionalitäten wird im nachfolgenden Abschnitt detailliert eingegangen.

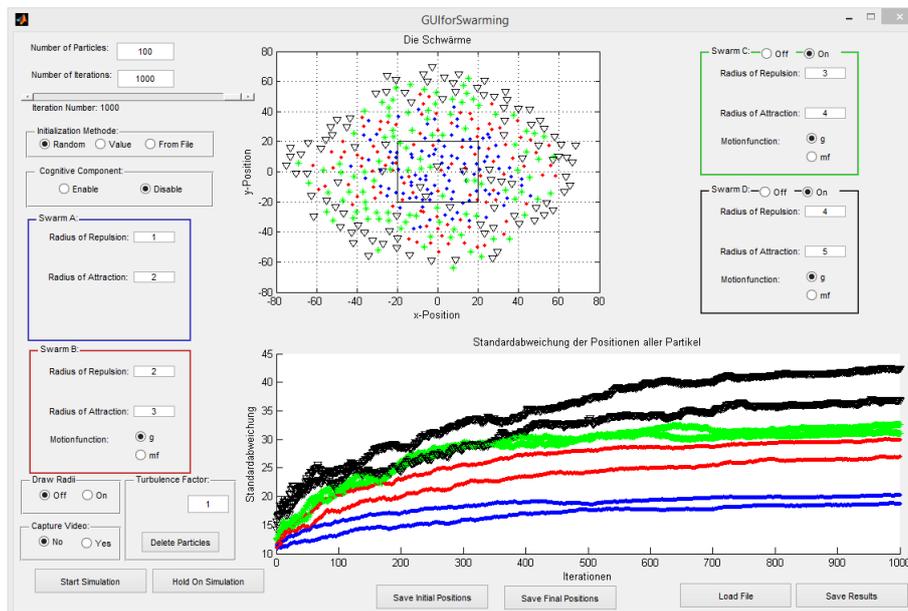
### 4.3 Gestaltung der Benutzeroberfläche

Die Benutzeroberfläche kann, aufgrund ihrer speziellen Ausrichtung auf die Forschung, nur mit Vorwissen im Bereich der Schwarmintelligenz verstanden und bedient werden. Aber es ist durchaus möglich, dass sie nicht ausschließlich von reinen Informatikern genutzt wird. Die grafische Benutzeroberfläche sollte daher nach Möglichkeit relativ einfach und gut verständlich gestaltet sein. Daher wurde bei Erstellen der grafischen Benutzeroberfläche darauf geachtet, dass viele der Empfehlungen aus den Windows User Experience Interaction Guidelines [UXGuide, 2014] eingearbeitet wurden. Es sollte jedoch auch bedacht werden, dass es sich hier lediglich um einen Prototyp handelt. Die entwickelte grafische Benutzeroberfläche stellt aus diesem Grund nur einen Designvorschlag dar, welcher noch beliebig abgewandelt werden kann. Abbildung 4.1 zeigt den Designvorschlag der grafischen Benutzeroberfläche, so wie sie in der aktuellen Version des Simulationstools verwendet wird. Im Nachfolgenden Teilabschnitt wird detailliert auf die einzelnen Elemente dieser grafischen Benutzeroberfläche eingegangen.

#### 4.3.1 Zweck und Verwendung der GUI-Elemente

Die entwickelte GUI bietet viele unterschiedliche Bedienelemente, welche zur detaillierten und individuellen Konfiguration der verschiedenen Simulationsparameter dienen.

So kann im linken oberen Edittext Feld die Anzahl der Partikel pro Schwarm festgelegt werden. Das dazugehörige Label „*Number of Particles:*“ soll es für



**Abbildung 4.1:** Die gesamte grafische Benutzeroberfläche des Prototypen mit vier simulierten Schwärmen

den Nutzer leichter verständlich machen, was an dieser Stelle einzugeben ist. Um es in der einer möglichen zukünftigen Version der GUI noch intuitiver für den Nutzer zu gestalten könnte die Abänderung dieses Labels in „*Number of Particles per Swarm:*“ in Betracht gezogen werden. Dadurch würde sofort klar werden, dass die eingegebene Zahl, die Partikelanzahl für alle Schwärme der Simulation definiert.

Darunter befindet sich ein Edittext Feld in dem der Nutzer die Anzahl der Iterationsschritte, über welche die Simulation laufen soll, eingeben kann. Auch hier verdeutlicht das zugehörige Label „*Number of Iterations:*“ den Zweck dieses Eingabefeldes.

Das darunter liegende Bedienelement ist ein Slider. In dessen Label wird die Nummer des aktuellen Iterationsschrittes angezeigt. Er dient nach Beendigung der Simulation als Werkzeug um beliebig durch alle durchlaufenen Iterationsschritten zu navigieren. Der ausgewählt Iterationsschritt wird dann interaktiv nicht nur im Label des Silder, sondern auch im oben mittig dargestellten Plot angezeigt. Auf den soeben erwähnten Plot wird im späteren Verlauf dieses Teilabschnitts noch detailliert eingegangen.

Ein Panel mit drei Radiobuttons ist das vierte Bedienelement auf der linken Seite der GUI. Hier kann der Nutzer durch seine Auswahl die Methode festlegen, mit der die Partikel für die Simulation initialisiert werden. Es besteht Möglichkeit einer zufällige Initialisierung (*Random*), einer Initialisierung mit fest definierten Werten (*Value*), diese initialisiert alle Partikel auf ein senkrechten Linie durch den Ursprung und einer Initialisierung mit Werten aus einer Datei mit entsprechenden Format, in der Aktuelle Version simples .txt-Format(*From File*).

Unterhalb liegt ein weiteres Panel, mit diesem kann der Nutzer bestimmen, ob während der Simulation die automatische Stucked Partikel Erkennung und

Behandlung deaktiviert oder aktiviert sein soll. Diese automatische Erkennung und Behandlung von Stucked Partikeln wird hier kurz als kognitive Komponente der Simulation bezeichnet. Diese Bezeichnung rührt aus der einprogrammierten Eigenschaft der Partikel, alle ihrer bisher eingenommenen Positionen abzuspeichern, sobald dieser Parameter aktiviert ist.

Im darunter befindlichen blau umrahmten Panel können der Repulsionradius sowie der Attractionradius des Schwarms A von Nutzer individuelle festgelegt werden. Alle Panels auf der GUI, in denen die Eigenschaften der Schwärme selbst festgelegt werden können, haben eine jeweilig spezifische Farbumrandung. Diese Farbumrandung entspricht der Einfärbung der Partikel des Schwarms in der Späteren Darstellung der Simulation. Die Übereinstimmung der Panelfarbe mit der Farbe der zugehörigen Partikel sorgt dafür, dass der Nutzer die Partikel im Simulationsplot intuitiver den einzelnen Schwärmen zuordnen kann.

Ein weiteres Panel zum Einstellen der Eigenschaften eines Schwarms befindet sich direkt unter dem blau gerahmten Panel. In diesem Panel kann der Nutzer die den Repulsion- sowie den Attractionradius des Schwarms B festlegen. Es ist durch eine rote Umrahmung gekennzeichnet. Anders als im vorherigen Panel kann der Nutzer hier noch zusätzlich die Funktion wählen, welche das Verhalten der Partikel des Schwarms bestimmt. Hierfür sind die beiden unteren Radiobuttons vorgesehen. Die Option die Bewegungsfunktion individuell festlegen zu können ist hier allerdings nur sehr minimal implementiert und dient eher als Platzhalter für zukünftige Versionen. Es sind ausschließlich zwei unterschiedlich gedehnte Versionen der begrenzten Attraction und Repulsion Funktion vorhanden, wobei „*g*“ die bei allen Simulationen in dieser Arbeit genutzte Funktion, und „*m.f*“ die leicht gedehnte Variante dieser Funktion, darstellt. In zukünftigen Versionen der GUI ist eine Erweiterung um mehr Radiobutton für viel unterschiedliche Bewegungsfunktionen oder möglicherweise sogar ein Feld individuellen Eingabe einer beliebigen Bewegungsfunktion denkbar.

Das Panel mit der Bezeichnung „*Draw Radii*“ ist ausschließlich ein Platzhalter. Mit den Radiobuttons kann der Nutzer in einer möglichen zukünftigen Version der GUI festlegen, ob das Simulationstool die Repulsion- und Attractionradien um die Partikel im Simulationsplot zeichnen soll oder nicht. Dadurch ließe sich die Änderung der Ausdehnung der Radien, bei aktivierte kognitiver Komponente, zur Laufzeit direkt sichtbar machen.

Im darunter befindlichen Panel kann der Nutzer festlegen ob ein Video von der Simulation erstellt werden soll oder nicht. Ist diese Komponente aktiviert wird während der Simulation eine auf mp4-Format komprimierte Videodatei erstellt, die ein Bild pro durchlaufenen Iterationsschritt enthält.

Daneben stellt die GUI ein Panel mit dem der Nutzer einen sogenannten Turbulenzfaktor in seine Simulation einfügen kann. Dies geschieht indem der Nutzer nach Ende der aktuellen Simulation eine Anzahl zufällig zu löschender Partikel eingibt und dann den Button „*Delete Particles*“ klickt. Dadurch wird die eingegebene Anzahl an Partikel, zufällig aus jedem Schwarm gelöscht und die Simulation mit der oben eingegebenen Anzahl von Iterationsschritten weitergeführt.

Der Button „*Start Simulation*“, welcher in der unteren linken Ecke der GUI zu finden dient, seinem Label entsprechend, dem starten der Simulation, mit den vorher eingegebenen Parametern.

Der daneben befindliche Button mit dem Label „*Hold On Simulation*“ dient zum pausieren einer aktuell durchgeführten Simulation im gerade durchlaufenen

Iterationsschritt.

Im unteren Bereich der GUI befindet sich eine Reihe weiterer Buttons. Der Button „*Save Initial Positions*“ dient zum abspeichern der initialen Positionen der Partikel der aktuellen Simulation in eine txt-Datei. Diese Datei kann später zum Beispiel bei einer erneuten Simulation bei der Initialisierung von einer Datei genutzt werden.

Mit dem Button „*Save Final Positions*“ können die Positionen der Partikel im letzten Iterationsschritt der Simulation in einer txt-Datei gespeichert werden. Auch diese könnten später zum Beispiel bei der Initialisierung von einer Datei genutzt werden, um beispielsweise eine noch unfertige Simulation fortführen zu können.

Der Button „*Load File*“ dient zum einladen ein kompletten bereits früher durchgeführten und abgespeicherten Simulation. Das Abspeichern ein vollständig durchgeführten Simulation, die dann später eingeladen werden kann, ist mit dem daneben liegenden Button mit dem Label „*Save Results*“ möglich.

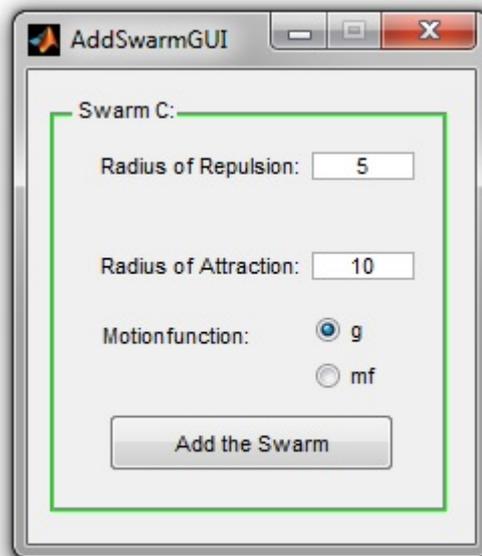
Im rechten oberen Bereich der GUI befinden sich zwei weitere Panels, diese dienen der Ergänzung von Schwärmen zu einer Simulation. Bis zu vier unterschiedliche Schwärme sind in einer Simulation möglich. Beide Panels im oberen Bereich Radiobuttons mit denen der Nutzer den entsprechenden Schwarm zu beziehungsweise abschalten kann. Alle weiteren Bedienelemente entsprechen denen des Panels für Schwarm B, die oben bereits beschrieben wurden.

Im mittleren Bereich der GUI befinden sich zwei Plots, die das Herzstück der grafischen Benutzeroberfläche dieses Simulationstools bilden. Der obere Plot stellt die simulierten Schwarmpartikel in ihrer Raumposition zu jedem Iterationsschritt grafisch dar. Dadurch kann der Nutzer Abläufe während jeder Simulation immer direkt betrachten.

Der darunter befindliche Plot stellt für jeden der Schwärme innerhalb der Simulation die Standardabweichung der Partikelpositionen, sowohl in x-, als auch in y-Richtung, zu jedem Iterationsschritt dar. Dadurch kann der Nutzer wesentlich intuitiver nachvollziehen wann, ob und in welchen Bereichen sich die einzelnen Schwärme ungefähr stabilisieren.

### 4.3.2 Ein erwähnenswertes verworfenes Gestaltungselement der grafischen Benutzeroberfläche

Im Verlauf der Entwicklung des Simulationstools hat sich auch die grafische Benutzeroberfläche immer wieder verändert, dabei gab es Ansätze die nicht in der aktuellen Version der Benutzeroberfläche des Prototypen vorhanden sind. Der wichtigste davon soll hier im Detail Erwähnung finden. Es ist eine kleine Pop-up GUI zum hinzufügen weiterer Schwärme in eine Simulation. Diese ist in Abbildung 4.2 dargestellt. Da die Möglichkeit gegeben werden sollte weitere Schwärme zu einer Simulation einzufügen, gab es anfänglich die Idee eines *AddNewSwarm* - Buttons. Beim anklicken dieses Buttons wurde eine kleine Pop-up GUI geöffnet, welche es ermöglichte die Größen des Repulsion- und Attractionsradius des neuen Schwarms individuell festzulegen. Außerdem bot das Pop-up noch einen Platzhalter, welcher es später möglich machen sollte den Partikeln des neuen Schwarms andere Bewegungsfunktionen zuzuordnen. Die Idee für diese Art der Schwarmergänzung kam auf, da es anfangs plausibel erschien die Option zu geben, unendlich viele weitere Schwärme einzufügen. Diese angedachte Funktionalität wurde allerdings im weiteren Verlauf der Arbeit ver-



**Abbildung 4.2:** Die Verworfenne GUI zum hinzufügen eines neuen Schwarms

worfen. Der Hauptgrund dafür liegt darin, dass es für diese Arbeit nachteilig gewesen wäre eine größere Anzahl ( $> 4$ ) an Schwärmen zu simulieren. Da dies der Übersichtlichkeit geschadet hätte. Ein weiterer Grund lag in der unerwartet hohen Komplexität, das hinzufügen unendlich vieler Schwärme über die GUI, unter der Verwendung von MatLab, zu ermöglichen. MatLab selbst stellt sich als eher ungeeignet dafür heraus. Die Folge ist, dass sich für eine statische Implementierung direkt auf der eigentlichen GUI entschieden wurde, auch wenn diese nur das Starten einer Simulation mit höchstens vier Schwärmen ermöglicht. Insgesamt könnte diese verworfene GUI Erweiterung allerdings zumindest von der Grundidee her für zukünftige Arbeiten in diesem Bereich von großem Nutzen sein.

# Kapitel 5

## Evaluierung

Dieses Kapitel widmet sich ausschließlich der Aufschlüsselung und Auswertung der aus der Simulation, der neu aufgestellten Annahmen und Definitionen, gewonnenen Ergebnisse. Dabei werden die unterschiedlichen Kombinationsmöglichkeiten der verschiedenen Parameter für Schwärme ausgetestet. Es werden die Resultat, der damit erstellten Simulationen dargestellt und schließlich Schlussfolgerungen daraus gezogen. Zusätzlich wird auf die, bei den Simulationen entstehenden, Instabilitäten und deren Behandlung eingegangen.

### 5.1 Wechselwirkungen

Beim Betrachten der Wechselwirkungen zwischen Schwärmen innerhalb einer Simulation wurden verschieden Parameterkombinationen gewählt, die allesamt andere Auswirkungen auf das Endergebnis der Simulation haben. Diese Kombinationen der Parameter orientiert sich an der grundlegenden Definition von Wechselwirkungen zwischen Schwärmen aus Kapitel 3 und versucht alle hieraus entstehenden Permutationen abzudecken. Die gewählten Parameterkombinationen werden in Tabelle 5.1 aufgelistet.

Jede der Simulationen wurde mit vier Schwärmen, welche alle jeweils 100 Partikel beinhalten, durchgeführt. Die Simulationen liefen allesamt über 1000 Iterationen. Jeder Schwarm erhielt eine eigene Farb- und Formkodierung zur besseren Unterscheidbarkeit. Die Partikel von Schwarm A sind mit blauen Punkten dargestellt. Die Partikel von Schwarm B sind rote Punkte. Partikel die zu Schwarm C gehören sind mit grünen Kreuzen verdeutlicht. Schwarze unausgefüllte Dreiecke stellen Partikel dar, welche zu Schwarm D gehören. Eine automatische Methode zur Behandlung von Stucked Partikel wurde bei alle in diesem Abschnitt aufgeführten Simulationen nicht verwendet. Das bedeutet, dass in einigen der dargestellten Simulationsresultate Stucked Partikel vorhanden sein können.

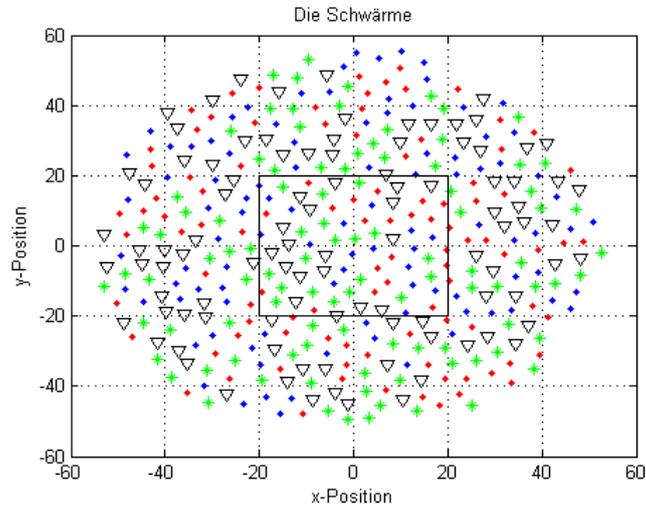
In Simulation 1 wurde eine Parameterkombination gewählt, bei der die Partikel aller Schwärme vollkommen identische Repulsion- und Attractionradien besitzen. So hat der Repulsionradius aller Schwärme den Wert 2 und der Attractionradius aller Schwärme den Wert 4. Abbildung 5.1 zeigt das Ergebnis der Simulation. Man erkennt, dass sich die Partikel aller Schwärme vollkommen homogen vermischt im gesamten Ausbreitungsgebiet verteilt haben. Die

Simulation	Schwarm	Repulsionradius	Attractionradius
1	A	2	4
1	B	2	4
1	C	2	4
1	D	2	4
2	A	2	3
2	B	2	4
2	C	2	5
2	D	2	6
3	A	2	4
3	B	3	4.5
3	C	5	7
3	D	6	7.5
4	A	1	5
4	B	2	5
4	C	3	5
4	D	4	5
5	A	1	2
5	B	2	4
5	C	4	6
5	D	6	7

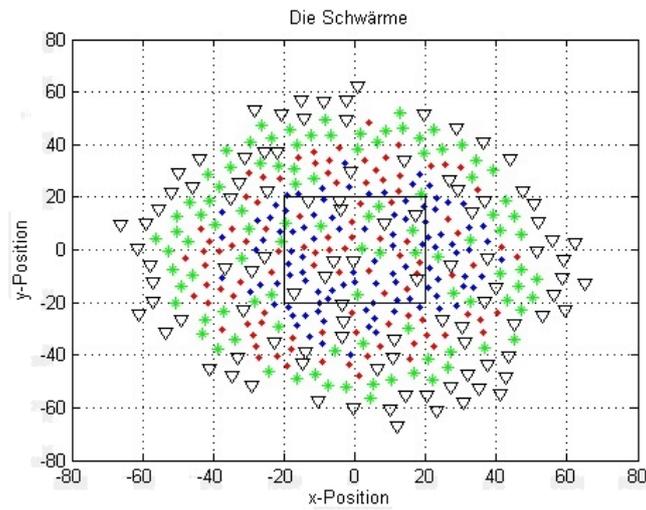
**Tabelle 5.1:** Die Tabelle listet die Parameterkombinationen der Schwärme in den fünf verschiedenen Simulationsläufen zu den Wechselwirkungen von Schwärmen auf

Schwärme waren dabei am Ende der Simulation vollkommen stabil. Diese Stabilität trat schon nach 231 Iterationen ein. Sie verhielten sich wie vermutet im Zusammenspiel wie ein einziger Schwarm und so entsprechend der Erkenntnisse aus [Blum, 2013]. Stucked Partikel traten, wie erwartet nicht auf.

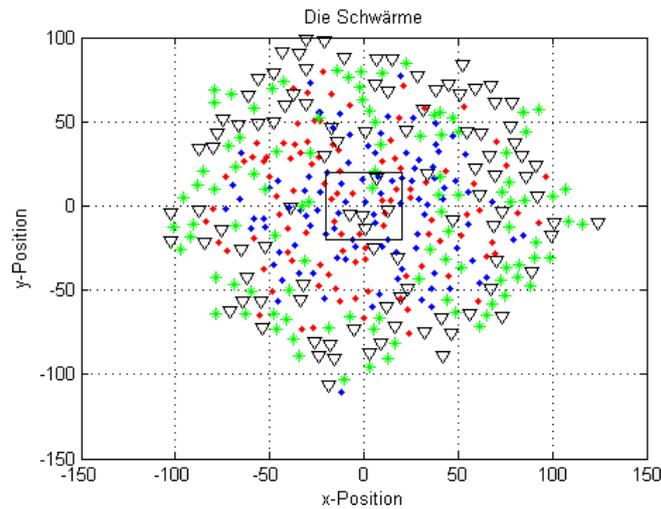
Für die zweite Simulation wurde für alle vier Schwärme ein Repulsionradius von 2 gewählt. Die Attractionradien unterscheiden sich voneinander. So hat Schwarm A einen Attractionradius von 3, Schwarm B besitzt einen Attractionradius von 4, für Schwarm C würde der Attractionradius auf einen Wert von 5 festgelegt und Schwarm D hat einen Attractionradius von 6. Das Resultat dieser Simulation ist in Abbildung 5.2 festgehalten. Es wird deutlich, dass sich die Partikel ihr Attractionradius entsprechend anordnen. Die Partikel des Schwarms A sind vor Allem im mittleren Bereich positioniert, jedoch gar nicht in an den äußeren Rändern. Dieses Verhalt ist darauf zurückzuführen, dass die Partikel des Schwarms A den kleinsten Attractionradius besitzen. Je größer der Attractionradius wird, umso weiter können die Partikel nach außen drängen. Aus diesem Grund sind die meisten Partikel am Außenrand Schwarm D zugeordnet, da dieser mit dem Wert 6 den größten Attractionradius besitzt. Da die Repulsionradien aller Schwärme identisch sind, könnten sich Partikel aller Schwärme theoretisch überall anordnen. Die Partikel mit größeren Attractionradius drängen nur weiter nach außen, weil diese durch ihren größeren Attractionradius eine größere Bewegungsfreiheit besitzen und so sich auch ihren Attractionradius entsprechend weiter außen anordnen können. Es sind, wie in der Abbildung zu erkennen, Ausreißer vorhanden, doch die meisten Partikel



**Abbildung 5.1:** Simulationsergebnis von vier Schwärmen mit identischen Eigenschaften



**Abbildung 5.2:** Simulationsergebnis von vier Schwärmen mit identischen Repulsionsradien und unterschiedlichen Attraktionsradien

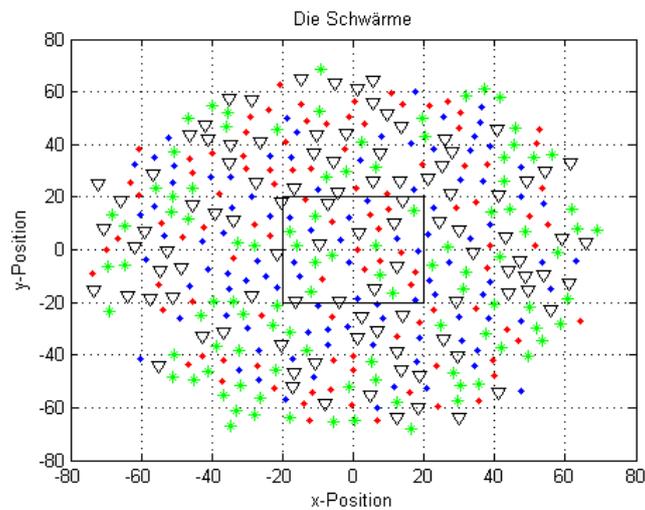


**Abbildung 5.3:** Simulationsergebnis von vier Schwärmen, wobei sich jeweils zwei der in ihren Repulsion- und Attractionradius überschneiden

ordnen sich automatisch nach dem Prinzip des von Innen nach Außen aufsteigenden Attractionradius an. Im hier dargestellten Ergebnis am tausendsten Iterationsschritt liegt allerdings keine hundertprozentige Stabilität vor, da Stucked Partikel vorhanden sind.

Die dritte Simulation arbeitet mit vier Schwärmen, von denen sich jeweils zwei in ihren Repulsion- und Attractionradien überschneiden. Schwarm A hat einen Repulsionradius von 2 und einen Attractionradius von 4. Schwarm B hat einen Repulsionradius von 3 und einen Attractionradius von 4,5. Zwischen Schwarm A und B liegt also eine Überschneidung vor. Die Parameter für Schwarm C sind ein Repulsionradius von 5 und ein Attractionradius von 7. Schwarm D hat einen Repulsionradius von 6 und einen Attractionradius von 7,5. Auch zwischen Schwarm C und D liegt eine Überschneidung vor. Abbildung 5.3 zeigt das Ergebnis der Simulation mit mit den oben genannten Parametern. Man sieht, dass diese Überschneidungen eine Auswirkung auf das Endresultat der Simulation haben. So sind im inneren Bereich des Ausdehnungsgebietes, mit Ausnahme einiger Ausreißer, hauptsächlich blaue und rote Partikel, also Partikel von Schwarm A und B, vorhanden. Wobei die roten Partikel des Schwarms B auch noch etwas weiter außen angeordnet sind. Der Grund hierfür ist der etwas größere Attractionradius von Schwarm B im Verhältnis zu Schwarm A. In den äußeren Bereichen des Ausdehnungsgebietes sind hauptsächlich grüne Partikel des Schwarms C und schwarze Partikel des Schwarms D zu finden. Wobei auch hier die schwarzen Partikel des Schwarms D hauptsächlich eine etwas weiter außen gelegene Position einnehmen als die des Schwarms C. In diesem Fall liegt der Grund ebenfalls darin, dass Schwarm D einen etwas größeren Attractionradius besitzt als Schwarm C. Die Partikel der Schwärme A und B vermischen sich allerdings kaum mit den Partikeln der Schwärme C und D.

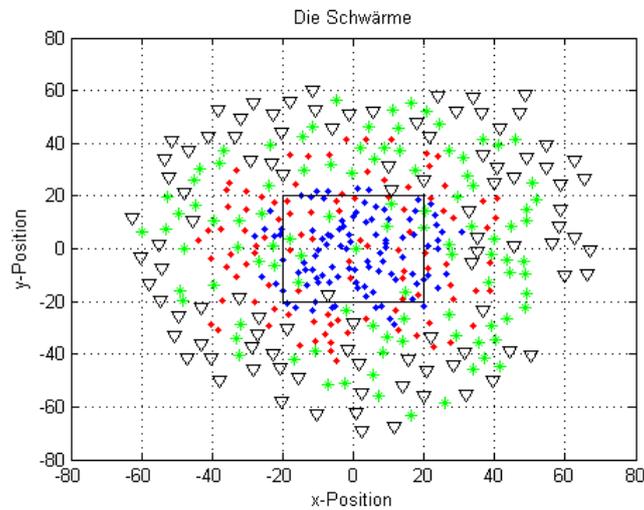
Für die vierte Simulation würde für alle Schwärme ein identischer Attractionradius von 5 festgelegt. Die Repulsionradien unterscheiden sich. Schwarm A



**Abbildung 5.4:** Simulationsergebnis von vier Schwärmen mit unterschiedlichen Repulsionradien und identischen Attractionradien

hat einen Repulsionradius von 1, Schwarm B einen Repulsionradius von 2, für Schwarm C würde ein Repulsionradius von 3 festgelegt und für Schwarm D ein Repulsionradius von 4. Das Endergebnis der Simulation mit diesen Parametern ist in Abbildung 5.4 dargestellt. Man erkennt, dass die Partikel aller Schwärme, wie auch bei der ersten Simulation, vollkommen homogen vermischt über das gesamte Ausdehnungsgebiet verteilt sind. Auch hier trat nach einer gewissen Anzahl von Iterationen vollständige Stabilität ein. (*in diesem Fall 342 Iterationen*) Am Ende der Simulation waren die Schwärme also ebenfalls vollkommen stabil. Die Schwärme verhielten sich in dieser Simulation unerwarteter Weise wie für einen einzigen Schwarm in [Blum, 2013] beschrieben. Insgesamt weist diese Simulation viel Ähnlichkeit zur ersten Simulation, bei der alle Repulsion- und Attractionradien identischen waren, auf. Lediglich die das Eintreten der vollkommenen Stabilität findet bei der vierten Simulation später statt als bei der ersten. Dieses Ergebnis lässt darauf schließen, dass Attractionradien eine wesentlich größere Rolle bei der Entstehung von Instabilität spielen, als Repulsionradien.

Bei der fünften und letzten Simulation wurden Repulsion- und Attractionradien gewählt, welche sich für alle Schwärme unterscheiden. Außerdem sind alle Repulsion- und Attractionradien vollkommen voneinander getrennt, Überschneiden gibt es nicht. Die Werte wurden für die Simulation wie folgt festgelegt. Schwarm A hat einen Repulsionradius von 1 und einen Attractionradius von 2. Für Schwarm B wurde ein Repulsionradius von 2 und ein Attractionradius von 4 eingestellt. Der Repulsionradius für Schwarm C wurde auf 4 festgelegt und sein Attractionradius auf 6. Schwarm D hat einen Repulsionradius von 6 und einen Attractionradius von 7. Das Resultat zur tausendsten Iteration der Simulation ist in Abbildung 5.5 festgehalten. In der Abbildung ist gut erkennbar, dass es trotz vereinzelter Ausreißern Hauptausdehnungsbereiche der Partikel jedes Schwarms gibt. So sind die blauen Partikel des Schwarms A in der Mitte verteilt, da diese den kleinsten Repulsion- und Attractionradius besitzen. Die roten Par-



**Abbildung 5.5:** Simulationsergebnis von vier Schwärmen mit unterschiedlichen Repulsion- und Attraktionsradien, die aufsteigend angeordnet sind

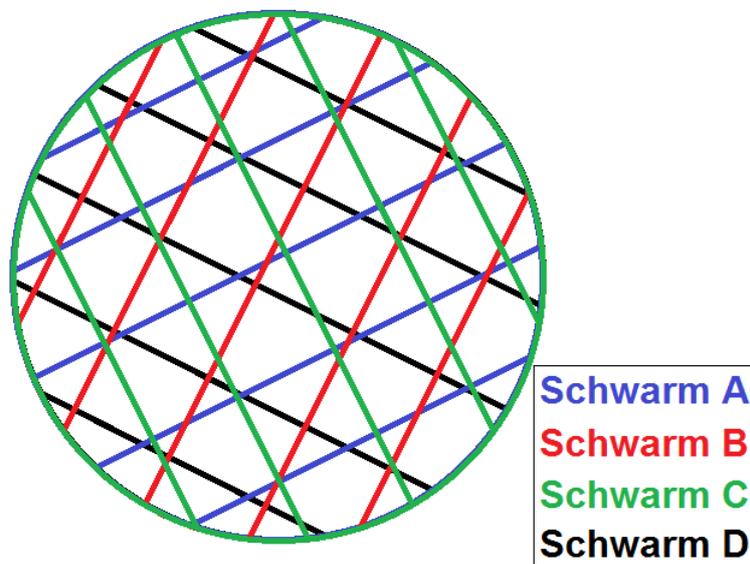
tikel des Schwarms B sind fast ringartig darum angeordnet. Ihr Repulsion- und Attraktionsradius ist etwas größer. Es folgen die grünen Partikel des Schwarms C. Deren Repulsion- und Attraktionsradius ist noch größer. Diese Partikel nehmen ebenfalls eine fast ringartige Verteilung an. Das Selbe gilt für die schwarzen Partikel des Schwarms D, die sich hauptsächlich am äußeren Rand des gesamten Ausdehnungsgebietes befinden. Der Grund dafür liegt darin, dass diese Partikel den größten Repulsion- und Attraktionsradius haben.

Bei allen der fünf durchgeführten Simulationen entstanden bestimmte Muster, welche immer wieder so oder ähnlich bei der Verwendung gleich skalierten Parameterkonstellationen auftreten. Die Erfassung dieser entstehenden Muster stellt eine der Hauptkenntnisse dieser Masterarbeit dar. Aus diesem Grund beschäftigt sich der folgende Abschnitt ausschließlich mit diesen Mustern.

### 5.1.1 Musterentstehung

Zur besseren Verdeutlichung der Muster, welche beim Wechselwirken mehrere Schwärme mit bestimmten Eigenschaften entstehen, werden die vier markantesten Muster in diesem Unterabschnitt noch einmal skizziert und beschrieben. Dabei wird auch die mathematische Beweisführung angetreten, dass diese Muster bei der Durchführung der Simulation auch wirklich entstehen. Dies geschieht unter Zuhilfenahme realer aus Simulationen gewonnener Daten. Diese sind die Standardabweichung der Positionen der Partikel in x-, sowie in y-Richtung und, wenn zum eindeutigen Beweis und Abgrenzung nötig, die Dichte der Partikel in bestimmten Raumbereichen. Die vier hier beschriebenen Muster erheben keineswegs den Anspruch auch Vollständigkeit, sondern stellen lediglich vier Muster dar, welche sich während mehrerer Simulationsdurchläufe mit bestimmten Parametern immer wieder ergeben haben.

- **Vollständige Vermischung**

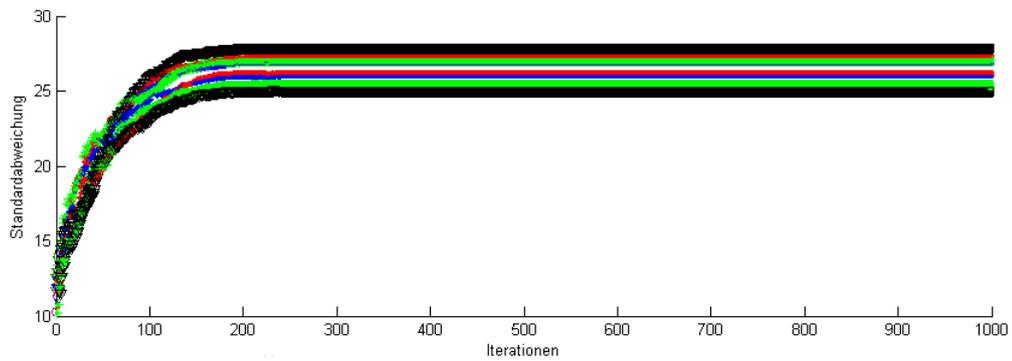


**Abbildung 5.6:** Das entstehende Anordnungsmuster bei Schwärmen mit identischen Repulsion- und Attractionradien. [Repulsionradius (A) = Repulsionradius(B) = Repulsionradius(C) = Repulsionradius(D) && Attractionradius (A) = Attractionradius(B) = Attractionradius(C) = Attractionradius(D)]

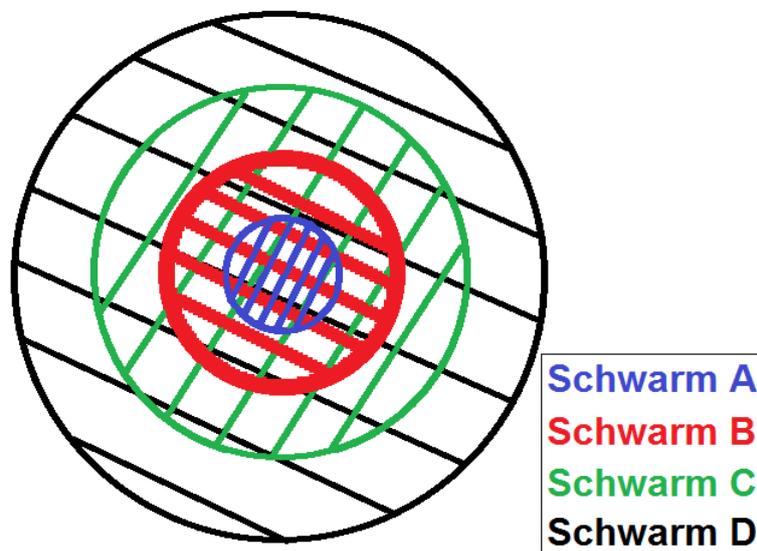
Je nach Wahl der Ausgangsparameter und der initialen Positionen der Partikel kann es im Ergebnis der Simulation zu einer Vermischung der aller Schwärme kommen. Das bedeutet, dass keiner der Schwärme eine eindeutige Position im Raum einnimmt, also ein Gebiet in dem die meisten seiner Partikel befinden. Solch eine vollständige Vermischung ist in Abbildung 5.x skizziert. Es ist gut erkennbar, dass sich die Partikel aller Schwärme über die gesamte Ausdehnung hinweg befinden. Ein solches Muster entsteht vor Allem wenn die Repulsion- und Attractionradien aller Schwärme identisch gewählt werden und die Initialisierung der Partikel in zufälliger, allgemeiner Lage erfolgt. Die in Abbildung 5.x dargestellte Standardabweichung der Partikelpositionen machen die Verteilung der Partikel aller Schwärme über den gesamten Ausdehnungsraum zusätzlich deutlich. So liegen die Funktionen für alle Schwärme sehr nah beieinander. Dies bedeutet das alle Schwärme die ungefähr gleiche Ausdehnung im Raum und damit gleiche Verteilung aufweisen müssen. Dies wird durch das Endresultat der Simulation, welches bereits in Abbildung 5.1 gezeigt wurde, ebenfalls verdeutlicht. Auch das Endresultat der vierten Simulation, welche in Abbildung 5.4 dargestellt ist, weist dieses Muster auf.

- **Vollkommen überlappendes Zwiebschalen Muster**

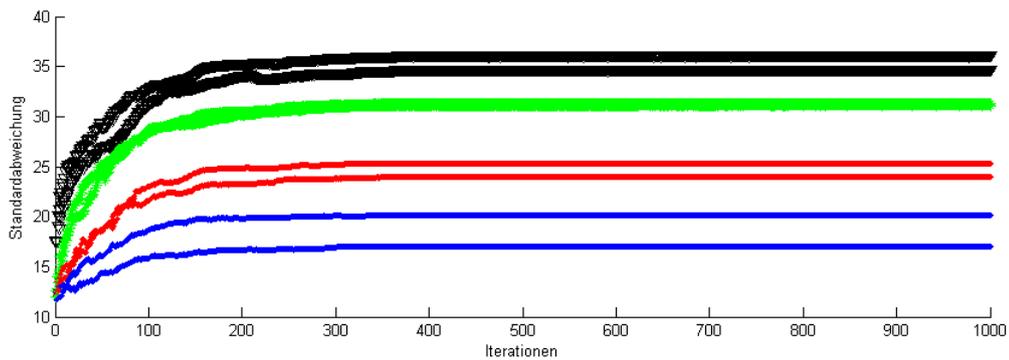
Im Fall des vollkommen überlappenden Zwiebschalen Musters ordnet sich der Großteil der Partikel der Schwärme in jeweils begrenzten Bereichen an. Dabei überschneiden sich die Bereiche, wobei sich ihre Größen unterscheiden. Abbildung 5.x skizziert dieses Muster. Es wird deutlich das ein Muster entsteht bei dem die Partikel des Schwarm A auf den klein-



**Abbildung 5.7:** Der Plot zeigt die Standardabweichung der Positionen der Schwarmpartikel zu jedem Iterationsschritt, bei einer vollständigen Vermischung. Alle Funktionen liegen sehr nah beieinander



**Abbildung 5.8:** Das entstehende Anordnungsmuster bei Schwärmen mit identischen Repulsionradien und aufsteigenden Attractionradien. [Repulsionradius (A) = Repulsionradius(B) = Repulsionradius(C) = Repulsionradius(D) && Attractionradius (A) < Attractionradius(B) < Attractionradius(C) < Attractionradius(D)]

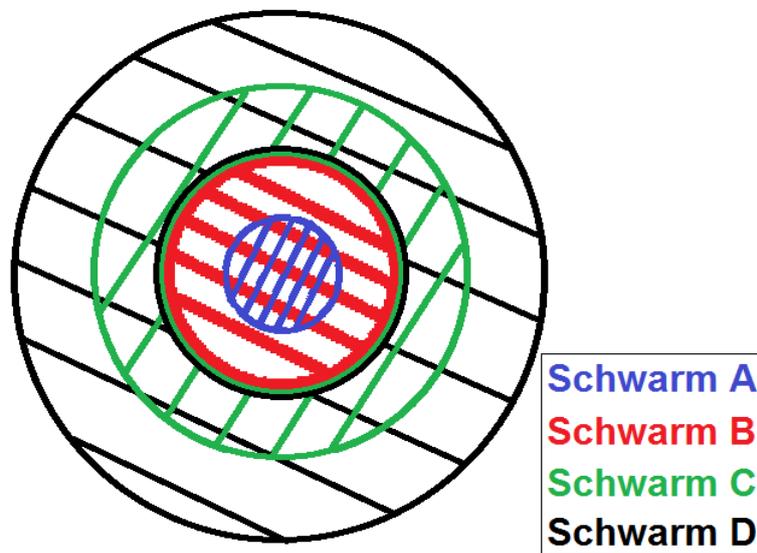


**Abbildung 5.9:** Der Plot zeigt die Standardabweichung der Positionen der Schwarmpartikel zu jedem Iterationsschritt, bei einem vollkommen überlappenden Zwiebelschalen Muster. Alle Funktionen der einzelnen Schwärme sind eindeutig weit voneinander entfernt

sten Raumbereich in der Mitte verteilt sind. Die Partikel des Schwarm B auf einen etwas größeren Raumbereich, der den Bereich des Schwarms A beinhaltet und so weiter. Ein solches Muster entsteht zumeist wenn alle Schwärme identische Repulsionradien und unterschiedliche Attraktionradien besitzen. Dabei ist es in den meisten realen Fällen zwar nicht so deutlich herausgebildet wie in Abbildung 5.x, allerdings bei genauer Betrachtung erkennbar und mathematisch nachweisbar. Der mathematische Nachweis kann leicht über die Standardabweichung der Positionen der Partikel der simulierten Schwärme erbracht werden. Diese Standardabweichung ist in Abbildung 5.x dargestellt. Hier sieht man, dass die Funktionen der Standardabweichung, welche zu den einzelnen Schwärmen gehören spätestens ab dem 200. Zeitschritt eindeutig voneinander getrennt sind. Dies stellt einen deutlichen Unterschied zu den Funktionsverläufen bei einer vollständigen Vermischung dar. Die beiden unteren blau dargestellten Funktionen können klar den Partikeln des Schwarms A zugeordnet werden, welche den kleinsten Ausdehnungsradius aufweisen, da sie den geringsten Attraktionradius besitzen. Die darüber liegenden beiden roten Funktionen sind präzise Schwarm B zuzuordnen. Dieser besitzt den zweitkleinsten Attraktionradius und damit die zweitkleinste Ausdehnung. Den nächstgrößeren Attraktionradius hat der grün dargestellte Schwarm C und damit hat dieser Schwarm auch die nächstgrößere Ausdehnung. Schwarm D hat die größte Ausdehnung und damit auch die am höchsten liegenden schwarzen Funktionen, weil er den größten Attraktionradius besitzt. Das zugehörige Endergebnis der Simulation wurde bereits in Abbildung 5.x dargestellt. Vor dem gerade genannten Hintergrund ist das vollkommen überlappende Zwiebelschalen Muster klar in der Abbildung zu erkennen.

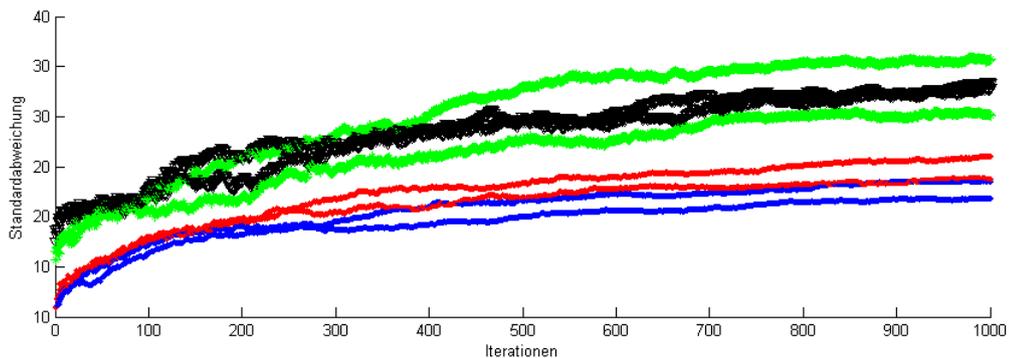
- **Paarweise disjunkt überlappendes Zwiebelschalen Muster**

Beim paarweise Disjunkten Zwiebelschalen Muster überlappen sich Ausdehnungsgebiete verschiedener Schwärme. Wobei die Überlappung in die-



**Abbildung 5.10:** Das entstehende Anordnungsmuster bei Schwärmen mit jeweils zwei sich, in Repulsion- und Attractionradien, überschneidenden Schwärmen.

sem Fall sich nicht auf Schwärme bezieht, sondern sich immer jeweils nur Gruppen von Schwärmen überlappen. Die Gruppen selber sind voneinander getrennt. Die Skizze in Abbildung 5.x macht diesen Sachverhalt deutlich. Hier sind vier Schwärme vorhanden, von denen sich jeweils nur zwei überlappen. In der Skizze überschneiden sich die Ausdehnungsgebiete von Schwarm A und B und die von Schwarm C und D. Dabei sind die Gebiete von A und B vollkommen von denen von C und D getrennt, hier gibt es keine Überschneidung. Solch ein Muster entsteht, wenn sich einige Schwärme in ihren Repulsion- und Attractionradien überschneiden andere jedoch nicht. Der in der Skizze dargestellt Idealfall tritt in der Realität allerdings eher selten auf. Meistens befinden sich noch sehr wenige Partikel anderer Schwärme in Bereichen, welche eigentlich nicht ihrem Ausdehnungsgebiet entsprechen. Dies kann man auch im Simulationsergebnis, welches bereits in Abbildung 5.3 dargestellt ist, erkennen. Im groben ist jedoch diese Muster. Den mathematischen Beweis hierfür liefert die in Abbildung 5.x gezeigte Standardabweichung der Positionen der Partikel und die in Abbildung 5.x dargestellte Dichteverteilung der Partikel im Raum. Die Graph der Standardabweichungen zeigt eine klare Trennung zwischen den zu Schwarm A und B gehörenden Funktionen und den zu Schwarm C und D gehörenden Funktionen. Wobei sich die Funktionen von Schwarm A und B teilweise überlagern, was auch für die Funktionen der Schwärme C und D gilt. Diese Überlagerungen verdeutlicht gut den Fakt, dass sich auch die Größe der Ausdehnungsräume der Schwärme ähnelt. Betrachte man zusätzlich die Dichteverteilungsfunktionen in Abbildung 5.x erkennt man auch hier eine Überlagerung von Schwarm A und B im Bereich von 30 bis 60. Im Fall der Schwärme C und D ist diese Überlagerung im Bereich von 60 bis 90 zu finden. Dies zeigt außerdem sehr gut, dass sich die Posi-

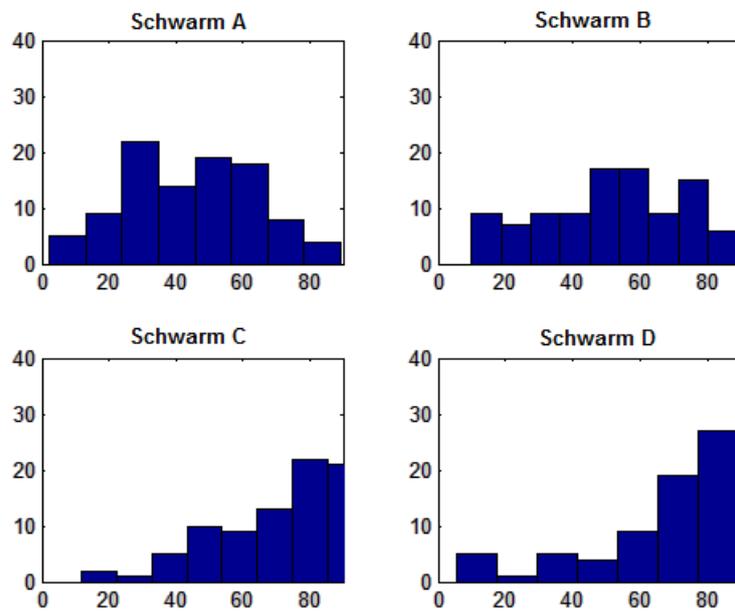


**Abbildung 5.11:** Der Plot zeigt die Standardabweichung der Positionen der Schwarmpartikel zu jedem Iterationsschritt, bei einem paarweise disjunkt überlappenden Zwiebelschalen Muster.

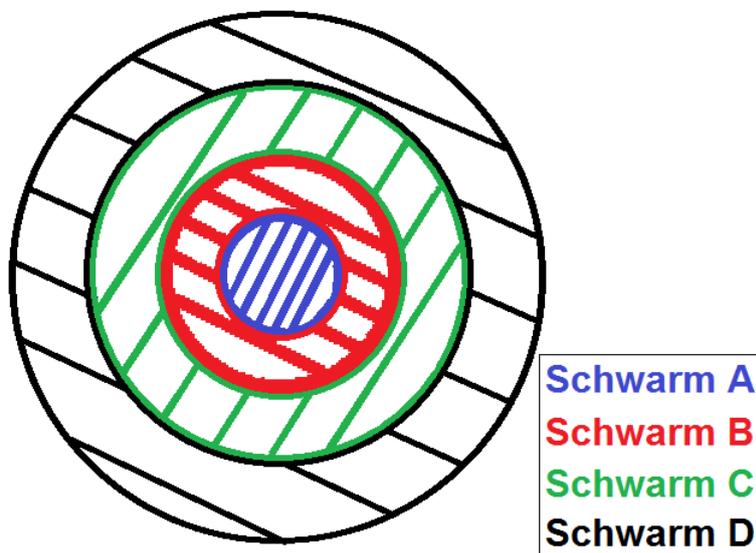
tionen an denen die meisten Partikel der Schwärme A und B verteilt sind, nicht mit den Positionen, an denen die meisten Partikel der Schwärme C und D verteilt sind, überschneiden. Die Verteilungsbereiche sind also zum größten Teil disjunkt.

### • Disjunktes Zwiebelschalen Muster

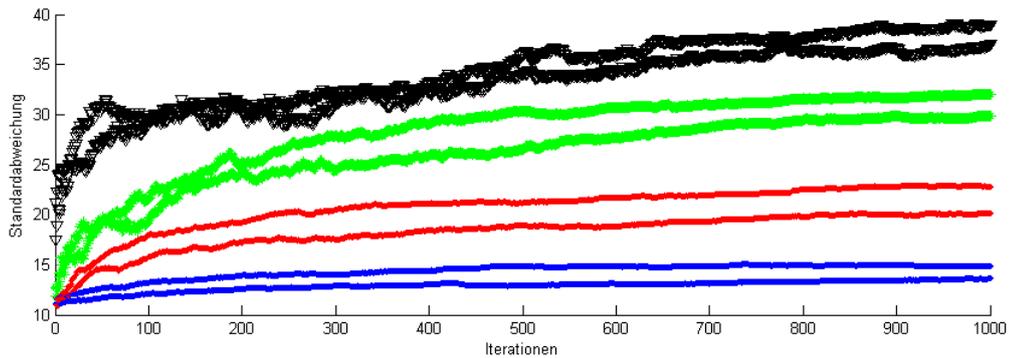
Ein disjunktes Zwiebelschalen Muster entsteht, wenn sich die Repulsion- und Attractionradien von Schwärmen nicht überschneiden. Ein solches Muster ist in Abbildung 5.x skizziert. Jeder Schwarm hat einen eigenen Ausdehnungsbereich innerhalb des Ausdehnungsbereichs aller Schwärme. Die Schwärme sind dabei, bis auf den Innersten mit dem kleinsten Repulsion- und Attractionradius, in einer Torus- beziehungsweise ringartigen Form um den Mittelpunkt angeordnet. Auch wenn es in der Realität zumeist nicht zu solch einer optimalen Anordnung kommt, wie in der Skizze dargestellt, so kann trotzdem mit mathematischen Mitteln bewiesen werden, dass so ein Muster grundlegend wirklich entsteht. In Abbildung 5.x sind die Funktionen der Standardabweichung der Positionen der Partikel jedes Schwarms dargestellt. Abbildung 5.x zeigt die Verteilung der Partikel jedes Schwarms im Raum. Beide Abbildungen aus der Simulation, deren Endresultat bereits in Abbildung 5.5 zu sehen ist, gewonnen. In der Abbildung des Endresultats sieht man, dass es in der Realität häufiger dazu kommen kann, dass sich ein Partikel weiter innen positioniert, dessen Schwarm eigentlich einen weiter außen liegenden Hauptverteilungsbereich besitzt. Doch geschieht dies nur mit verhältnismässig wenigen Partikel, weshalb das disjunkte Zwiebelschalen Muster trotzdem entsteht. Dies gut an den Funktionen der Standardabweichungen zu erkennen. Diese sind spätestens ab der 500. Iteration eindeutig voneinander getrennt. Was klar eine von Schwarm A zu D größer werdende Standardabweichung der Partikelpositionen beweist. Das dies alleine allerdings noch nicht genügt um Eigenschaft zu beweisen, dass die Schwärme disjunkt verteilt sind muss die Dichteverteilung der Schwarmpartikel der Betrachtung hinzugefügt werden. Hierbei wird eindeutig belegt, dass die Partikel des Schwarms



**Abbildung 5.12:** Der Plot zeigt die Dichteverteilung der Schwarmpartikel im Endresultat, bei einem paarweise disjunkt überlappenden Zwiebelschalen Muster.



**Abbildung 5.13:** Das entstehende Anordnungsmuster bei Schwärmen mit aufsteigenden Repulsion- und Attractionradien. [Repulsionradius (A) < Repulsionradius(B) < Repulsionradius(C) < Repulsionradius(D) && Attractionradius (A) < Attractionradius(B) < Attractionradius(C) < Attractionradius(D)]



**Abbildung 5.14:** Der Plot zeigt die Standardabweichung der Positionen der Schwarm-partikel zu jedem Iterationsschritt, bei einem disjunkten Zwiebelschalen Muster.

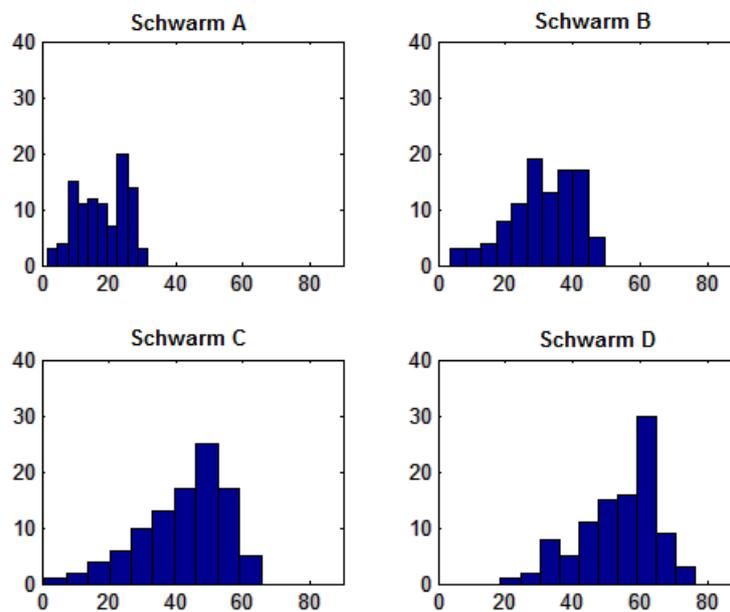
A hauptsächlich im Bereich 10 bis 30, die Partikel des Schwarms B im Bereich 30 bis 40, die des Schwarms C in Bereich 40 bis 50 und die des Schwarms D im Bereich 60 bis 70 verteilt sind. Dadurch ist bewiesen, dass die Partikel hauptsächlich disjunkt in eigenen Bereichen verteilt sind. Das disjunkte Zwiebelschalen Muster existiert und ist mit diesem Vorwissen auch klar in der Darstellung der Endergebnis der Simulation erkennbar.

## 5.2 Instabilität durch Wechselwirkungen

Durch das Wechselwirken von Schwärmen mit unterschiedlichen Eigenschaften entstehen nicht nur bestimmte Muster, sondern es kommt auch in vielen Fällen zu instabilen Verhalten einzelner Partikel. Diese Partikel mit instabilen Verhalten sind die in Kapitel 3 bereits detailliert besprochenen Stucked Partikel. Sie werden in diesem Abschnitt erneut aufgegriffen um ihr reales Verhalten innerhalb von Simulationen zu untersuchen, da sie der Hauptgrund die Instabilität vom Schwärmen beim Wechselwirken vieler Schwärme mit unterschiedlichen Eigenschaften sind. Außerdem wird in diesem Abschnitt untersucht in wie weit das zufällige Löschen einzelner Partikel aus stabilen oder nahezu stabilen Schwarmkonstellationen zu Instabilität oder womöglich zu Stabilität führen kann.

### 5.2.1 Stucked Partikel

Bei jeder Simulation, bei der Schwärme mit unterschiedlichen Eigenschaften in Wechselwirkung treten, kommt es zu Instabilität durch Stucked Partikel. Durch die Simulationen wurde klar, dass diese Partikel sich in einem bestimmten kleinen Gebiet des Raumes in zufälliger Weise bewegen. Diese Bewegung wird endlos fortgeführt, solange keine Behandlungsmaßnahmen ergriffen werden und die gesamte Umgebung des Partikels stabil bleibt. Stabil bleiben bedeutet in diesem Fall, dass eine feste Gruppe von stabil im Raum stehenden Partikeln existiert, deren Stabilität nicht von den Stucked Partikeln beeinträchtigt wird. Um das Phänomen der Stucked Partikel noch etwas detaillierter Untersuchen zu können, wurde das zufällige Bewegungsmuster eines Stucked Partikels in einem



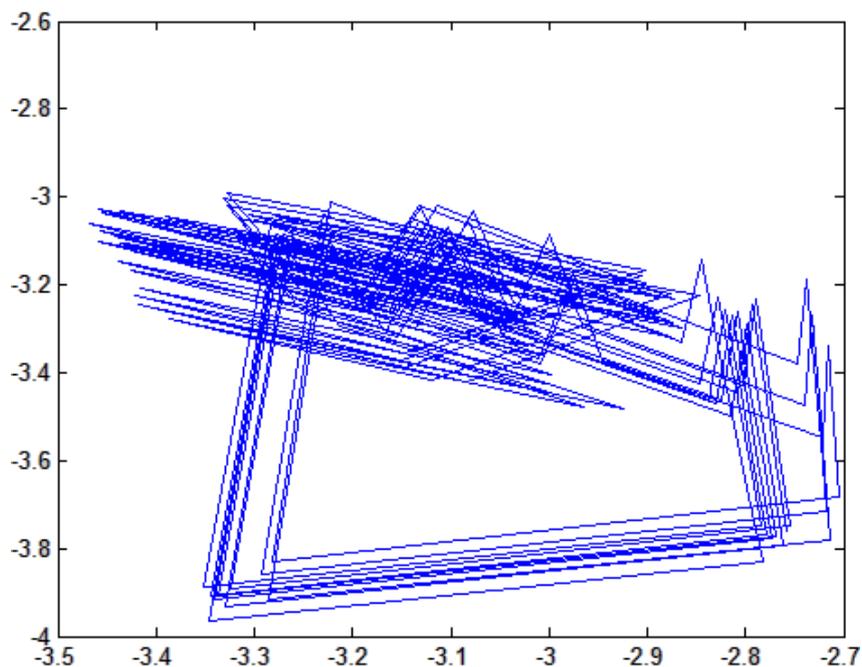
**Abbildung 5.15:** Der Plot zeigt die Dichteverteilung der Schwarmpartikel im Endresultat, bei einem disjunkten Zwiebelschalen Muster.

Diagramm über 200 Iterationen hinweg als Trajektorie festgehalten. Diese ist in Abbildung 5.x zu sehen. Die in der Abbildung dargestellte Trajektorie macht deutlich, dass die Bewegung offensichtlich doch keinem vollkommen zufälligen Muster folgt. Es entsteht eine... . Diese Unerwartete Erkenntnis über Stucked Partikel, führt zu der logischen Konsequenz, dass Stucked Partikel bei weitem Komplexer sind, als hier bisher angenommen. Aus diesem Grund, sollte auf diesem Ansatz in zukünftigen Arbeiten aufgebaut werden. Eine detailliertere Untersuchung dieser Erkenntnis hätte allerdings leider den Rahmen dieser Arbeit gesprengt. Da hier noch weitere allgemeinere Betrachtungen durchgeführt wurden.

### 5.2.2 Zufälliges Löschen von Partikeln

Um eine möglichst vollständige Betrachtung durchzuführen, wurde auch ein Turbulenz Faktor untersucht. Dieser Turbulenz Faktor, ist die Möglichkeit der Entstehung von Turbulenzen und damit Instabilität, durch das zufällige Löschen einzelner Partikel. Dabei werden die Partikel aus einer stabilen oder nahezu stabilen Konstellation von vier Schwärmen gelöscht. Abbildung 5.x zeigt eine nahezu stabile Konstellation und die daraus durch Löschen von Partikeln entstandenen Konstellationen. Außerdem die jeweils zu jeder Konstellation zugehörigen Funktionen der Standardabweichungen über 1000 Iterationen dargestellt.

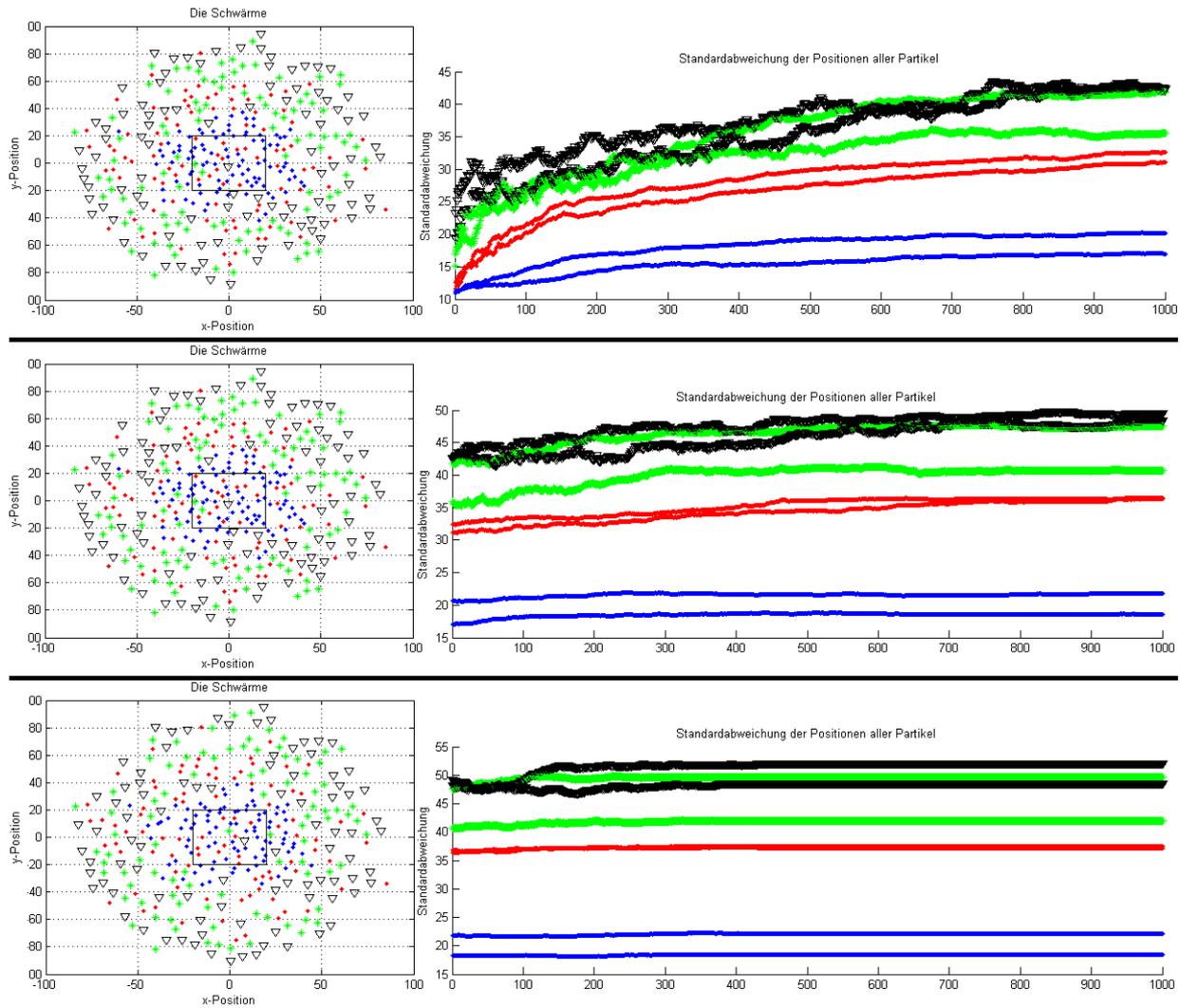
Die obere Darstellung zeigt das Endresultat, nach einer Simulation über 1000 Iterationen, mit vier Schwärmen, a 100 Partikel, welche sich alle in ihren Eigenschaften unterscheiden. Die gewählten Eigenschaften sind in Tabelle 5.2 festgehalten. Diese Eigenschaften sind für alle weiteren Simulationen gleich.



**Abbildung 5.16:** Die Trajektorie eines Stucked Partikels über zweihundert Iterationen hinweg

In der oberen Darstellung ist eine annähernd stabile Konstellation zu sehen. Dies ist auch in den Funktionen der Standardabweichung zu erkennen. Bei der mittleren Abbildung wurden zufällig fünf Partikel pro Schwarm gelöscht. Die dadurch, nach 1000 Iterationen, entstehende Konstellation wird noch etwas stabiler. Die Standardabweichung der Positionen der Partikel vergrößert sich jedoch auch. Auch hier ist dieses Phänomen klar, im dazugehörigen Graph der Standardabweichungen, sichtbar. Die untere Darstellung zeigt, eine noch wesentlich stabilere Konstellation. Diese entstand 1000 Iterationen nach dem zufälligen Löschen von jeweils fünf weiteren Partikeln aus der mittleren Konstellation. Im Graph der Standardabweichungen ist erkennbar, dass die untere Konstellation sich nach ca. 450 Iterationen vollkommen stabilisiert hat.

Aus diesen Resultaten lassen sich zwei Dinge eindeutig schließen. Erstens durch Löschen von Partikeln aus Schwarmkonstellationen kommt es kurzzeitig zu stärkerer Instabilität. Zweiten diese Instabilität löst sich jedoch nach einigen Iterationen auf und die Konstellation stabilisiert sich noch stärker als zuvor. In der hier durchgeführten Simulation trat eine, klar erkennbare, vollständige Stabilisierung bereits nach dem zweiten Löschen von jeweils fünf Partikeln pro Schwarm ein. Die Schwärme bestehen nicht mehr, wie zu Beginn, aus 100 Partikeln pro Schwarm, sondern nur noch aus 90. Die hier alle 1000 Iterationen durchgeführte Löschung von fünf Partikeln pro Schwarm, sorgt allerdings für eine hundertprozentige Stabilisierung, nach etwas weniger als 2450 Iterationen. Die Erkenntnis, dass durch zufällige Löschung von Partikeln Stabilität entsteht, wirkt im ersten Moment unplausibel. Sie wird aber klar, wenn man



**Abbildung 5.17:** Der obere Graph zeigt eine nahezu stabile Konstellation von Partikelschwärmen nach 1000 Iterationen. Der mittlere Graph zeigt die gleiche Konstellation nach dem zufälligen löschen von jeweils fünf Partikeln pro Schwarm und weiteren 1000 Iterationen. Der untere Graph zeigt die Konstellation nach dem erneuten Löschen von jeweils fünf Partikeln. Nebenstehend sind jeweils die zugehörigen Funktionen der Standardabweichung zwischen den Partikelpositionen abgebildet

Schwarm	Repulsionradius	Attractionradius
A	1	2
B	2	4
C	4	6
D	6	7

**Tabelle 5.2:** Die Tabelle listet die Parameterkombinationen der Schwärme in der Simulation zum zufälligen Löschen von Partikeln auf

bedenkt, dass sich die meisten Partikel nach einigen Iterationen in weiter zueinander entfernte Position bewegen, als zu Beginn einer Simulation. Werden nun aus dieser distanzierten Stellung Partikel gelöscht, haben alle anderen Partikel mehr Platz. Durch die neu entstandenen Freiräume, können sich die Partikel noch besser zueinander positionieren. So wandern Partikel die vorher instabil (stucked) oder von instabilen Partikeln eingeengt waren, in die freiwerdenden Flächen. Dieser Fakt löst Stucked Partikel nach und nach auf und sorgt schließlich für mehr Stabilität. Außerdem ist durch das anfängliche Wandern einiger Partikel in Richtung der neuen Freiräume, auch der Grund für die kurzweilige größere Instabilität nach dem Löschen geklärt.

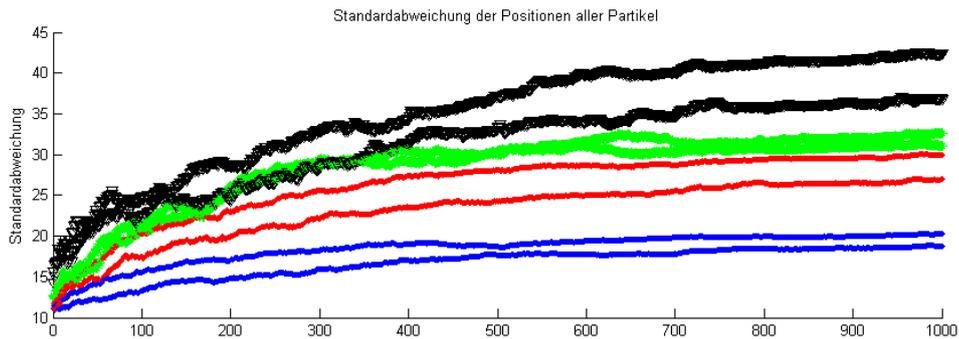
Aus den Ergebnissen kann man ableiten, dass es theoretisch möglich ist Multi-Schwarmkonstellationen auch durch das sukzessive Löschen von Partikeln zu stabilisieren. Für Multi-Schwarmkonstellationen, bei denen der Verlust einiger Partikel verschmerzbar ist, eine in Zukunft durchaus bedenkenswerte Alternative. Weitere Möglichkeiten zur Stabilisierung von instabilen Schwarmkonstellationen, werden im nachfolgenden Abschnitt detailliert ausgewertet.

### 5.3 Stabilisierung durch Behandlung von Stucked Partikeln

Mit Hilfe der kognitiven Komponente des Simulationstools kann der Instabilität entgegengewirkt werden. Um diese Annahme zu belegen wurden fünf Simulationen durchgeführt. Alle Simulationen haben die selbe Initialisierung und verwenden identische Simulationsparameter. Lediglich die Behandlungsmethode der Instabilität unterscheidet sich bei jeder Simulation. So ist die erste Simulation ohne Radiananpassung durchgeführt. Die zweite Simulation mit Nutzung von Festhalten zur Behandlung von Stucked Partikeln. Bei der dritten Simulation werden adaptive Radien mit festem Wert verwendet. Die vierte Simulation benutzt prozentuale Anpassung der Radien. Die Fünfte verwendet prozentuale Anpassung und die Beschleunigungserweiterung. Die Simulationsparameter sind aus Tabelle 5.3 ablesbar. Alle Simulationen wurden mit 100 Partikeln pro Schwarm und über 1000 Iterationen durchgeführt. Während der Durchführung der Simulationen wurde festgestellt, dass es keine signifikanten Unterschiede zwischen der Verwendung von Positionsänderung und Winkelmessung als Stucked Partikel Erkennungsmethoden gibt. So unterschied sich die Dauer, bis zum Eintreten der Stabilität, bei beiden nicht. Was für Erkennung durch Positionsänderung gilt, gilt ebenso für Erkennung durch Winkelmessung. Aus diesem Grund wird hier darauf verzichtet die Plots der Standardabweichungsfunktionen für

Schwarm	Repulsionradius	Attractionradius
A	1	2
B	2	3
C	3	4
D	4	5

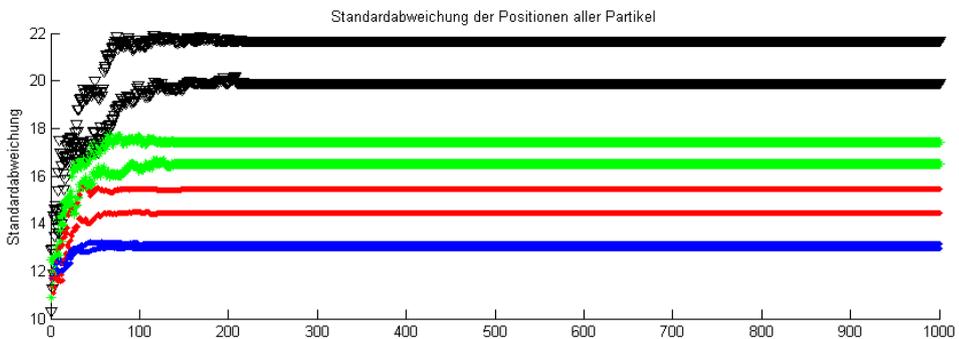
**Tabelle 5.3:** Die Tabelle listet die Parameterkombinationen der Schwärme in den Simulationen zur Stabilisierung durch Behandlung von Stucked Partikeln auf



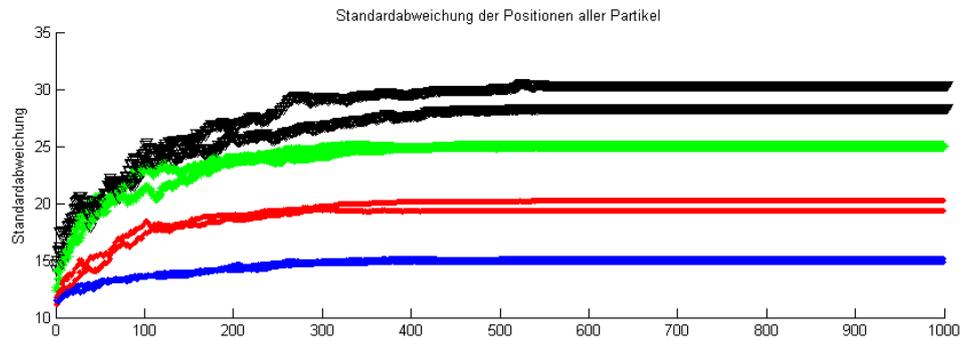
**Abbildung 5.18:** Die Funktionen der Standardabweichung der Partikelpositionen für eine Simulation ohne Radienanpassung

beide Erkennungsmethoden abzubilden und zu beschreiben. Die Betrachtung beschränkt sich hier ausschließlich die Erkennungsmethode durch Positionsänderung. Diese Betrachtung kann direkt auf die Erkennung durch Winkelmessung übertragen werden.

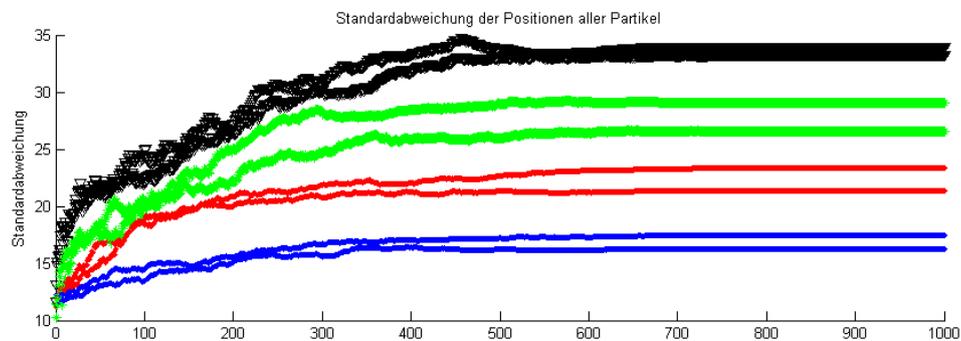
Die Funktionen der Standardabweichungen der ersten Simulation wurden in Abbildung 5.x festgehalten. Hier ist eindeutig erkennbar, dass ohne Verwendung einer Behandlungsmethode keinerlei Stabilität eintritt. Die Funktionen sind bis zur tausendsten Iteration noch sehr unruhig. Dies gilt für alle Schwärme. In der zweiten Simulation, welche in Abbildung 5.x dargestellt ist, wurde Festhalten zur Behandlung von Stucked Partikeln genutzt. In dieser Simulation wur-



**Abbildung 5.19:** Die Funktionen der Standardabweichung der Partikelpositionen für eine Simulation mit Stabilisierung durch Festhalten



**Abbildung 5.20:** Die Funktionen der Standardabweichung der Partikelpositionen für eine Simulation mit Radienanpassung durch einen festen Wert (*hier* 0,09)



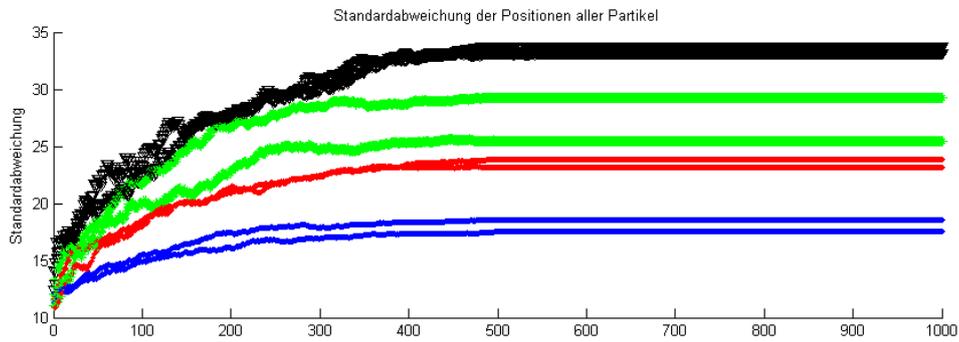
**Abbildung 5.21:** Die Funktionen der Standardabweichung der Partikelpositionen für eine Simulation mit prozentualer Radienanpassung (*alterRadius* \*0,99)

den die Radien jedes Partikels, welcher als *stucked* erkannt wurde, auf null gesetzt. Dadurch kam es zum sofortigen Verharren dieser Partikel an ihrer Position. So wurde bereits nach 220 Iterationen vollständige Stabilität erreicht. Der Beweis dafür ist, dass ab diesem Zeitpunkt die Funktionen konstant bleiben.

Die dritte Simulation wurde unter Verwendung einer Behandlungsmethode durch Radienanpassung mit einem festen Wert von 0,09 durchgeführt. Abbildung 5.x zeigt die Standardabweichungsfunktionen der Simulation. Man erkennt, dass die vollständige Stabilität später eintritt als in der zweiten Simulation. Der Grund hierfür liegt darin, dass der feste Wert mit 0,09 sehr klein gewählt ist. Dadurch wird die Anpassung genauer, aber auch langsamer.

Die vierte Simulation wurde mit einer prozentualen Radienanpassung, mit einem Wert von 0,99, als Behandlungsmethode der Instabilität durchgeführt. Diese Simulation ist in Abbildung 5.x veranschaulicht. Die Anzahl der Iterationen bis zum Eintreten vollständiger Stabilität ist hier noch etwas größer als in der vorherigen Simulation. Der Grund liegt hier ebenfalls darin, dass eine Radienanpassung mit 99% der vorherigen Radiengröße sehr genau ist. Sogar noch wesentlich genauer als die Anpassung in Simulation drei. Sie wird allerdings dadurch noch rechenaufwändiger und benötigt so noch mehr Iterationen.

Für Simulation fünf wurde die Behandlungsmethode der vorherigen Simulation um eine zusätzlich Beschleunigung der *Stucked* Partikel erweitert. Das Ergebnis ist im Plot der Abbildung 5.x festgehalten. Man erkennt deutlich, dass



**Abbildung 5.22:** Die Funktionen der Standardabweichung der Partikelpositionen für eine Simulation mit prozentualer Radienanpassung (*alterRadius* \*0,99) und Beschleunigung von Stucked Partikeln (*Positionänderung* \*5)

weniger Iteration bis zur vollständigen Stabilisierung benötigt werden. Dies ist der Beweis dafür, dass die Erweiterung eindeutig eine schnellere Stabilisierung ermöglicht. Die Erweiterung ist sinnvoll.

Die Simulationen beweisen, dass alle Behandlungsmethoden zur Stabilisierung eingesetzt werden können. Jeder der Methoden vermag es eine Multi-Schwarmkonstellation nicht nur leicht zu stabilisieren, sondern nach entsprechend vielen Iterationen sogar eine vollständige Stabilisierung herbeizuführen. Dieser Fakt, ist die Wichtigste Erkenntnis dieser Masterarbeit. Ohne die Verwendung einer Behandlungsmethode, bleibt die Instabilität bestehen. Wie schnell die vollständige Stabilisierung eintritt, hängt von der Genauigkeit der Methode ab. Je ungenauer eine Methode ist, desto geringer ist ihr Rechenaufwand, was zu einer schnellen Stabilisierung führt.

## Kapitel 6

# Zusammenfassung und Ausblick

Da die Forschung im Bereich der Schwarmintelligenz ein noch relativ junger Forschungsbereich ist, treten beim detaillierten Betrachten eines neuen Problems in diesem Bereich immer wieder weitere Problemfelder auf, die es zu lösen gilt. So auch bei dieser Arbeit. Aus diesem Grund werden in diesem Kapitel vier mögliche weiterführende Ansätze zusammengetragen. Zusätzlich wird abschließend diese Arbeit noch einmal kurz zusammengefasst.

### 6.1 Mögliche weiterführende Ansätze

Im Verlauf der Arbeit wurden eine Reihe von Verbesserungen, Erweiterungen und eventuell sinnvoller Umgestaltungen deutlich. Außerdem stellten sich Punkte heraus bei denen es möglicherweise sinnvoll wäre weiterführende Forschungsarbeit zu betreiben. Einige vielversprechende dieser weiterführenden Ansätze werden im Folgenden kurz angerissen:

- **Übersetzung der Implementierungen in andere Programmiersprachen**

Die bisher genutzte Programmierumgebung für die in dieser Arbeit durchgeführten Simulation ist MatLab. Mit ihr wurden sämtliche für diese Arbeit benötigten Implementierungen umgesetzt. Dies schließt die erstellte GUI genauso ein, wie auch die Schwärmverhaltensweisen der Partikel selbst, die entwickelten Detektions- und Behandlungsmaßnahmen und auch alle weiteren Funktionalitäten. MatLab stellte sich allerdings im Verlauf der Arbeit, als eher unpraktikabel zur Implementierung einiger der Funktionalität heraus. So konnten manche der erwünschten Funktionalitäten nur sehr schlecht, mit hohem Aufwand oder teilweise sogar gar nicht umgesetzt werden. Des Weiteren arbeitet MatLab bei der Berechnung einiger der genutzten Algorithmen und Abfragen eher ineffizient. Für eine verbesserte Performance und zukünftig bessere und möglicherweise leichtere Ergänzung von Funktionalitäten, ist es überlegenswert die für diese Arbeit durchgeführten Implementierungen in eine andere besser geeignete Programmiersprache, wie beispielsweise C#, zu übersetzen.

- **Verbesserung der zugrundeliegenden Algorithmen**

Alle für diese Arbeit entwickelten oder in diese Arbeit verwendeten Algorithmen, sind nicht vollständig auf ihre Performance optimiert. Der Grund hierfür liegt hauptsächlich darin, dass das Hauptaugenmerk dieser Arbeit auf der Untersuchung und Interpretierung der Verhaltensweisen beim Wechselwirken unterschiedlicher Schwärme liegt. Die Zeit hätte deshalb für eine Entwicklung schnellerer Methoden nicht ausgereicht. Weitere Untersuchungen im Bereich der Multi-Schwärme werden jedoch mitunter noch wesentlich rechenaufwändigere Simulationen erfordern, als die Simulationen in dieser Arbeit. Darum wäre es sinnvoll in der Zukunft an diesen Algorithmen zu arbeiten und an ihnen Verbesserungen vor Allem im Bezug auf ihre Performance vorzunehmen.

- **Umgestaltung und Verbesserung des GUI-Prototypen**

Für diese Arbeit wurde ein eigenes Simulationstools entwickelt, welche auch eine GUI beinhaltet. Diese GUI wurde eigentlich nur dafür entwickelt um die Simulationen für diese Arbeit schneller und übersichtlicher durchführen zu können. Es hat sich dann jedoch herausgestellt, dass die Idee eine GUI für Simulationen in diesem Bereich zu erstellen vollkommen neu ist. So könnte die entwickelte GUI auch als Designvorschlag für weitere GUIs dienen, die zukünftige Forschungsarbeiten in diesem Bereich unterstützen und erleichtern könnten. Darum ist es überlegenswert an der GUI selbst zukünftig Umgestaltungen oder Erweiterungen vorzunehmen, welche ihrer Verbesserung dienlich sind, da die GUI im Moment bei weitem noch nicht den Anspruch erhebt ausgereift zu sein.

- **Entwicklung weiterer neuer Methoden zur Stabilisierung von Schwärmen**

Da es beim Wechselwirken mehrerer Schwärme mit verschiedenen Eigenschaften häufig zu Instabilen Verhaltensweisen kommt, wurden für diese Arbeit eine Reihen von Methoden entwickelt, welche der Stabilisierung von Schwärmen dienen. Dieser erheben allerdings keineswegs den Anspruch auch Vollständigkeit. Hier besteht eventuell die Option weitere neue Methoden zu entwickeln. Außerdem beziehen sich alle entwickelten Methoden auf das Erkennen und behandeln von Sucked Partikeln. Es könnte aber möglicherweise noch vollkommen andere Herangehensweisen geben, durch welche eine Stabilisierung von instabilen Schwärmen ermöglicht werden kann. So könnte man sich beispielsweise auf den Schwarm als Ganzes beziehen und sich nicht nur auf einzelne Partikel beschränken. Oder man wendet sich Methoden zu, welche mit dem sukzessiven Löschen von einzelnen Partikeln zur Stabilisierung arbeiten. Solche Methoden könnten laut einiger Erkenntnissen dieser Arbeit funktionieren.

- **Untersuchung möglicher weiterer Muster und Verhaltensänderung bei der Wechselwirkung zwischen viel größerer Mengen unterschiedlicher Schwärme**

In dieser Arbeit wurden die Wechselwirkungen zwischen zwei bis vier Schwärmen untersucht. Auch wenn sich aus diesen Untersuchungen viele Aussagen auf das Verhalten bei der Wechselwirkung noch mehr Schwärme

übertragen lassen, ist es denkbar, dass sich beim Wechselwirken einer noch wesentlich größeren Anzahl von Schwärmen noch weitere oder veränderte Verhaltensweisen zeigen. Diese könnten sich zum Beispiel in der Entstehung neuer, ganz anderer, Muster oder Partikel Bewegungsformen äußern. Die Untersuchung der Wechselwirkungen zwischen Schwärmen ist also keineswegs mit vollständig durchgeführt oder geschweige denn, abgeschlossen. Aus diesem Grund sind weitere Betrachtungen in diesem Bereich durchaus sinnvoll und in der Zukunft definitiv in Erwägung zu ziehen. Dies gilt vor Allem für die Untersuchung der Wechselwirkungen zwischen einer großen Anzahl von mehr als 100 Schwärmen mit unterschiedlichen Eigenschaften. Hierbei sollte allerdings von Beginn an bedacht werden, dass dieses Vorhaben höchstwahrscheinlich sehr Performanceaufwändig sein wird.

### • Übertragung der Stabilisierungsmethoden auf reale Roboterschwärme

Eine der Hauptmotivationen dieser Masterarbeit war das leider häufig unvorhersehbare und dadurch instabile Verhalten realer Roboter in Schwärmen. Dieses wird unter Anderem durch äußere Einflüsse, kleinen Anomalien in der Bauweise der Roboter oder sonstigen unerwarteten Reaktionsweisen der Roboter ausgelöst. Die Roboter sorgen dadurch dafür, dass reale Roboterschwärme sich häufig sehr instabil verhalten. Die Übertragung und schlussendliche Anwendung der für dieser Arbeit entwickelten Stabilisierungsmethoden auf echte im Schwarm agierende Roboter, könnte dabei helfen die auftretenden Probleme in Zukunft zu reduzieren oder möglicherweise vollständig zu lösen. Eine Übertragung dieser Stabilisierungsmethoden auf reale Roboterschwärme erscheint daher als ein sinnvoller nächster Schritt, um die Forschung innerhalb dieses Bereiches voranzutreiben. So könnten die Methoden detailliert auf ihre Tauglichkeit in der Realität untersucht werden.

## 6.2 Zusammenfassung

Das Ziel dieser Arbeit war es, erste Untersuchungen der Wechselwirkungen zwischen Schwärmen mit unterschiedlichen Eigenschaften anzustellen. Dabei wurden viele neue Erkenntnisse gewonnen. Zu diesen Erkenntnissen gehört auch die Feststellung von Problemen, welche zu Instabilität führen. Aus dem Wissen über diese Probleme wurden zusätzlich Ansätze entwickelt sie zu beheben.

In der Arbeit selbst wurden dazu zuerst die Grundlagen erläutert, die den Grundstein für diese Arbeit legten. An dieser Stelle wurde Bezug auf die Begriffe der Schwärme, der Künstlichen Intelligenz und der Schwarmintelligenz genommen. Zusätzlich wurden die für diese Arbeit wichtigen Grundregeln, welche das Schwärmverhalten der Individuen eines Schwarms auslösen, intensiv erörtert.

Nachdem die für die Arbeit wichtigen Grundlagen aufgeschlüsselt wurden, konnten die neuen Ansätze bei der Betrachtung von vielen Schwärmen detailliert dargelegt werden. Dazu gehörten die erstmals definierten Arten von Wechselwirkungen zwischen Schwärmen, das Phänomen der Stucked Partikel und die möglichen Erkennungs- und Behandlungsmethoden für Stucked Partikel. Das Verständnis all dieser Ansätze, war wiederum sehr wichtig für weiteren Abschnitte der Arbeit.

Im Anschluss daran wurden die eigentliche Implementierung vorgestellt und wie sie, im für diese Arbeit entwickelten Simulationstool, umgesetzt wurden. Es wurden die wichtigen Funktionalitäten des Simulationstools und ihre Arbeitsweise erläutert. Dazu gehört der Algorithmus zur Berechnung des Schwarmverhaltens, welcher die Grundlage aller Simulationen dieser Arbeit bildet und die Realisierung der Erkennungs- und Behandlungsmethoden von Stucked Partikeln. Es wurden auch Pseudocodes zu besseren Veranschaulichung genutzt. Als Abschluss wurde auf die Gestaltung der Nutzeroberfläche des Simulationstools eingegangen. Dabei wurde der Zweck und die Verwendung jedes einzelnen Elements der Nutzoberfläche genau beschrieben. Außerdem wurde, mit der Pop-up GUI zum hinzufügen weitere Schwärme, auf ein möglicherweise zukünftig noch sehr wichtiges verworfenes Gestaltungselement der grafischen Benutzeroberfläche eingegangen.

Die Evaluierung der gewonnenen Erkenntnisse bildet das vorletzte Kapitel dieser Masterarbeit. Hierbei wurde intensiv auf die, aus verschiedenen Simulationsläufen gewonnen, Ergebnisse eingegangen. Die daraus gewonnenen Erkenntnisse, vor Allem im Bezug auf Musterentstehung wurden detailliert aufgeführt und deren Korrektheit bewiesen. Im Anschluss wurde das Problem der Instabilität ausgewertet. Zum Abschluss des Kapitels wurden die Möglichkeit der Stabilisierung von wechselwirkenden Schwärmen durch die Behandlung von Stucked Partikeln bewertet.

Insgesamt stellt diese Arbeit einen enormen Sprung, für die Forschung im Bereich der Schwarmintelligenz dar. Vor Allem im Bezug auf das Verständnis der entstehenden Strukturen und Verhaltensweisen beim aufeinandertreffen mehrere Schwärme. Diese wurden zuvor noch niemals untersucht. So wurden dabei viele neue Erkenntnisse gewonnen. Zu diesen gehören beispielsweise die Musterentstehung beim Zusammentreffen mehrere Schwärme oder die Erkenntnis über die dabei entstehende unerwünschte Instabilität. Diese war zwar beim Wechselwirken von Schwärmen mit unterschiedlichen Eigenschaften zu erwarten, deren Ausprägung war jedoch bisher nicht klar. In dieser Arbeit wurden sie nun erstmals bewusst sichtbar und so könnte das Phänomen der Stucked Partikel als Auslöser der Instabilität herausgestellt und neu definiert werden. Die Erkenntnisse über die Ausprägungen und Eigenschaften der Instabilität halfen in dieser Arbeit zusätzlich dabei, erste Ansätze zu entwickeln, um die Instabilität einzudämmen oder sogar vollkommen zu unterbinden. Das stellt einen großen Schritt und zusätzlich die wichtigste Erkenntnis dieser Arbeit dar. In Zukunft könnten auf Grundlage dieser Ansätze neue Methoden entwickelt werden, welche auch dem aktuellen Problem der Instabilität in realen Roboterschwärmen entgegenwirken könnten. Eine weitere Errungenschaft dieser Arbeit ist, die Idee der Erstellung und Nutzung einer GUI zum konfigurieren, durchführen und auswerten von Schwarmsimulationen. Durch diese könnten Simulationen im Bereich Schwarmintelligenzforschung in Zukunft einfacher und schneller durchführbar werden. Abschließend ist festzuhalten, dass diese Masterarbeit einen weiteren wertvollen Schritt in Richtung der Erforschung und Verwendung der Prinzipien der Schwarmintelligenz darstellt.

# Literaturverzeichnis

[Blum, 2013] Blum, Jessica: *Kontrolle von fliegenden Robotern mithilfe von Schwarmintelligenz*, 2013.

[Pescara, 2013] Pescara, Erik: *Beobachtung und Analyse dynamischer Systeme mit Hilfe fliegender Roboter*, 2013.

[TMW, 2014] TheMathWorks, Inc.: *MATLAB - Die Sprache für technische Berechnungen*, 2014.

[ITW, 2014] *Beitrag zu grafische Benutzeroberflächen auf ITWissen.info* .  
<http://www.itwissen.info/definition/lexikon/graphical-user-interface-GUI-Grafische-Benutzeroberflaeche.html>, Stand: 05.05.2014

[UXGuide, 2014] *Windows User Experience Interaction Guidelines*.  
<http://msdn.microsoft.com/en-us/library/windows/desktop/aa511258.aspx>,  
Stand: April 2014.

[LaIn, 2012] *Der LaTeX-Index: Eine Befehlsübersicht im World Wide Web*.  
<http://www.weinelt.de/latex/>, Stand: Juli-August 2012

[RoMa, 2007] Rozin, Valeri; Margaliot, Michael : *The Fuzzy Ant*. In: IEEE Computational Intelligence Magazine, 2007.

[Beni et al., 1989] Beni, Gerardo; Wang, Jing: *Swarm Intelligence in Cellular Robotic Systems*. In: Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, Seiten 26–30, Juni 1989.

[Gazi et al., 2011] Gazi, Veysel; Passino, Kevin M.: *Swarm Stability and Optimization*. Springer-Verlag, 2011.

[Liu et al., 2000] Liu, Yang und Kevin M Passino: *Swarm intelligence Literature overview*. Department of Electrical Engineering, the Ohio State University, März 2000.

- [Hinchey et al., 2007] Hinchey, Michael G.; Sterrit, Roy und Rouff, Chris: *Swarms and Swarm Intelligence*. In: IEEE Computer, 40 (4), Seiten 111 - 113, April 2007.
- [Spiteller 2013] Spiteller, Michael: *Das traditionelle Verständnis von Schwarmintelligenz*. 2013.
- [Lobo 2013] Lobo, Sascha: *Den Schwarm interessiert nur das Ergebnis*. In: Spiegel Online, abgerufen am 8. Mai 2014.
- [Dorigo et al. 2000] Dorigo, M.; Di Caro, G. und Stützle, T.: *Special issue on Ant Algorithms*. In: Future Generation Computer Systems, Volume 16, Nummer 8, 2000.
- [Bryant 1991] Bryant, Greg: *The Oregon Experiment after Twenty Years* In: RAIN, Winter/Spring 1991, Volume 14, Number 1, 1991.
- [Alexander 1975] Alexander, Christopher: *The Oregon experiment*. In: Center for Environmental Structure series 3, New York, NY : Oxford University, 1975.
- [Mostaghim, 2013] Mostaghim, Sanaz: *Abbildungen zur Vorlesung Swarm Intelligence an Otto von Guericke Universität, Magdeburg*, 2013.
- [UXGuide, 2012] *Windows User Experience Interaction Guidelines*.  
<http://msdn.microsoft.com/en-us/library/windows/desktop/aa511258.aspx>,  
Stand: Juni 2012.
- [Reynolds, 1987] Reynolds, Craig: *Flocks, herds and schools: A distributed behavioral model*. In: Computer Graphics, 21(4), Seiten 25-34, Juli 1987,  
<http://www.red.com/cwr/boids.html> , Stand: 17.05.2014
- [Reynolds, 1999] Reynolds, Craig: *Steering Behaviors For Autonomous Characters*. In: Conference Proceedings of the 1999 Game Developers Conference, Seiten 763-782, 1999.
- [WiPro, 2014] *Wikipedia Artikel 'Prototyping (Softwareentwicklung)'* .  
[http://de.wikipedia.org/wiki/Prototyping\(Softwareentwicklung\)](http://de.wikipedia.org/wiki/Prototyping(Softwareentwicklung)), Stand:  
17.05.2014